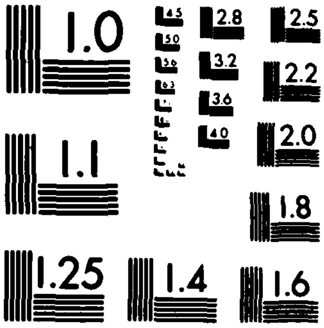
AD-A143 032 AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM (ACMS) VOLUME 1/3
2 PROGRAM DOCUMENTATION(U) CONSTRUCTION ENGINEERING
RESEARCH LAB (ARMY) CHAMPAIGN IL C E HERRING JUN 84
UNCLASSIFIED CERL-TR-P-158-VOL-2 F/G 9/2 NL


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

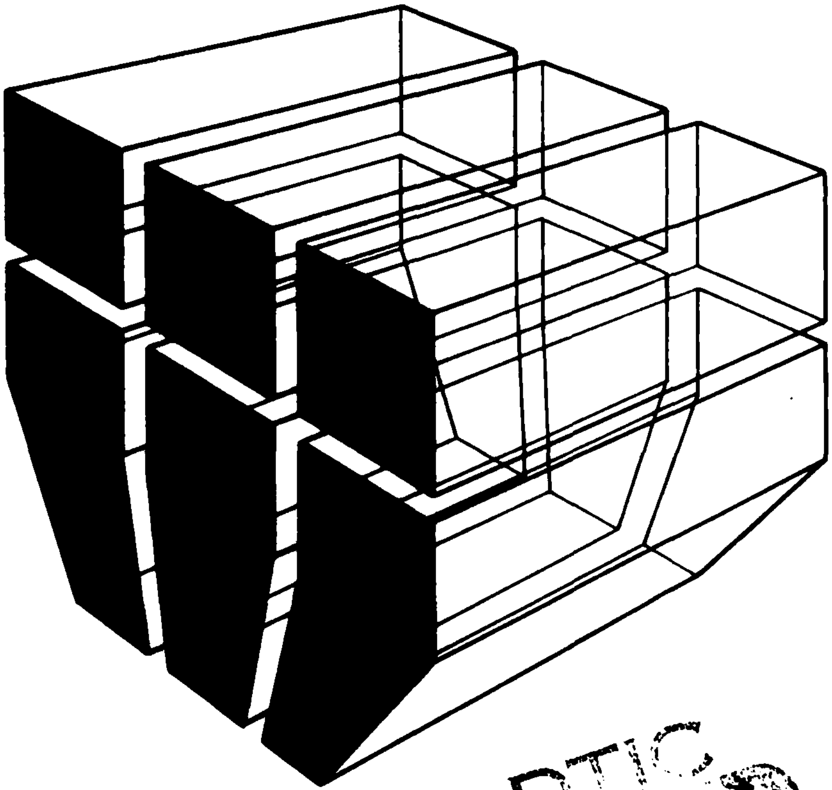
(2

CERL

US Army Corps
of Engineers
Construction Engineering
Research Laboratory

TECHNICAL REPORT P-158
June 1984

AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM (ACMS), VOLUME II: PROGRAM DOCUMENTATION



by
Charles E. Herring, Jr.

Approved for public release; distribution unlimited.

DISCLAIMER

The computer program described herein is furnished by the Government and is accepted and used by the recipient with the express understanding that the United States Government makes no warranties, expressed or implied, concerning the accuracy, completeness, reliability, useability, or suitability for any particular purpose of the information and data contained in this program or furnished in connection therewith, and the United States shall be under no liability whatsoever to any person by reason of any use made thereof. The program belongs to the Government. Therefore, the recipient further agrees not to assert any proprietary rights therein or to represent this program to anyone as other than a Government program.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER CERL-TR-P-158	2. GOVT ACCESSION NO. AD-A143032	3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM (ACMS), VOLUME II: PROGRAM DOCUMENTATION

5. TYPE OF REPORT & PERIOD COVERED
Final

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Charles E. Herring, Jr.

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
U.S. ARMY
CONSTRUCTION ENGINEERING RESEARCH LABORATORY
P.O. BOX 4005, CHAMPAIGN, IL 61820

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
4A762731AT41-E-038

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
June 1984

13. NUMBER OF PAGES
250

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Copies are available from the National Technical Information Service
Springfield, VA 22161

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Automated Construction Management System
ACMS
management information systems
microcomputers
construction

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Automated Construction Management System (ACMS) has been developed to manage Army Engineer unit construction projects. It is a full-feature project management system implemented on a multiuser microcomputer system. The approach in developing the system was to exploit commercial software programs and hardware to minimize programming requirements and thus allow researchers to spend more time finding uses for the system.

(Continued on next page)

Block 20. (Cont'd).

In FY83 the ACMS was fielded as a pilot system at the 18th Engineer Brigade in the Federal Republic of Germany. The pilot system was used to manage more than 100 construction tasks during 1983.

Volume I is a complete user's guide describing ACMS version 84.0 and the hardware and software as fielded at the 18th Engineer Brigade; it also gives detailed examples for using the modules and describes input requirements and output reports. Volume II contains the data base documentation for ACMS version 84.0, including system requirements, installation instructions, and program listings.

ACMS version 84.0 is available from the Command and Control Microcomputer User's Group (C^{2}MUG). Contact: Chief, CECOM SDSD, ATTN: DRSEL-FL-SDSD (C^{2}MUG), Fort Leavenworth, KS 66027. Telephone (913) 684-7550 (COMM), or 552-7550 (AUTOVON).

FOREWORD

This investigation was conducted for the Assistant Chief of Engineers, Office of the Chief of Engineers (OCE), under Project 4A762731AT41, "Design, Construction, and Operation and Maintenance Technology for Military Facilities"; Task E, "Military Engineering"; Work Unit 038, "Engineer Unit Microcomputer Applications." The applicable STO is 81-5.1:19. The OCE Technical Monitor was Dr. Clemens Meyer, DAEN-ZCM.

This work was performed by the Facilities Systems (FS) Division of the U.S. Army Construction Engineering Research Laboratory (USA-CERL). Jennifer Young was an associate investigator for the project, and Jane-Ping Chiu made significant contributions as a programmer. E. A. Lotz is Chief of USA-CERL-FS. U.S. Army 18th Engineer Brigade personnel directly involved in fielding the ACMS were COL Charles Williams, Commander; LTC Melvin Lynch, S-3; LTC Larry Izzo, S-3; MAJ Russell Baldwin, S-3, Operations; MAJ David Phillips, S-3, Plans; 1LT Lordes Goodnight, S-3, Operations; Ms. Julie Osborn, S-3, Operations; and SP4 Vicky McDaniels, S-3, Operations.

COL Paul J. Theuer is Commander and Director of USA-CERL, and Dr. L. R. Shaffer is Technical Director.

CONTENTS

AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM (ACMS),
VOLUME II: PROGRAM DOCUMENTATION

1 INTRODUCTION

Background

During FY83 the U.S. Army 18th Engineer Brigade at Grafenwoehr, Federal Republic of Germany (FRG), was engaged in a massive construction effort to upgrade training ranges so that modern weapon systems, such as the M-1 tank, could be used in exercises. With an estimated cost of $40 to 60 million and involvement of 4000 soldiers plus a civilian support group, the project needed a responsive, accurate construction management system.

The U.S. Army Construction Engineering Research Laboratory (USA-CERL) was asked to develop a microcomputer-based Automated Construction Management System (ACMS) for the 18th Engineer Brigade project. The pilot system was fielded at the 18th Engineer Brigade Headquarters in January 1983. The system's initial capability consisted of a Network Analysis Module and a General Applications Module with an electronic spreadsheet and word processing. In April 1983, USA-CERL added a Data Base Management Module, a Weekly Progress Report Module, and a Reports Generation Module.

Version 84.0 is the Data Base Management Module of the ACMS and was designed to meet the 18th Engineer Brigade's needs. It provides for computerized management of equipment and personnel by individual projects.

Purpose

The purpose of this report is to provide program documentation and a programmer's reference for ACMS Version 84.0.

Approach

To develop the ACMS, off-the-shelf hardware and software were exploited to minimize programming requirements and to enable researchers to spend more time designing system applications, described in Volume I. For this reason, the dBASE-II™ data base management program by Ashton-Tate, Inc., was used to implement the Data Base Management Module.

The approach in documenting the ACMS was to explain the system requirements, installation of data base programs, and data base structure, and to give complete program listings.

Mode of Technology Transfer

The ACMS has been given to the 2nd Engineer Group in Korea and the 20th Engineer Brigade at Fort Bragg, NC. The applications programs, written in dBASE-II, are available to all Army Engineer units from the U.S. Army Command and Control Microcomputer User's Group (C^2MUG).*

The products of this research will ultimately be interfaced with related Army initiatives and will be interfaced with other Army initiatives, such as the Military Computer Family.[1] The goal is to have a well-defined set of engineer-specific applications that can be implemented on military computer hardware and software by 1990.

*Contact: Chief, CECOM SDSD, ATTN: DRSEL-FL-SOSD (C^2MUG), Fort Leavenworth, KS 66027; telephone (913) 684-7550 (COMM), or 552-7550 (AUTOVON).

[1] H. J. Cashman, "Army Battlefield Automation--The Military Computer Family Concept," Signal (July 1983), pp 37-42.

2 ACMS DATA BASE DOCUMENTATION

System Requirements

ACMS version 84.0 is written in dBASE-II version 2.4 and consists of 7500 lines of code in 33 command files that occupy 290 Kbytes of storage. The two-user version has 51 data base and index files occupying 102 Kbytes, and the single-user version has 33 data base files occupying 66 Kbytes. The system will operate on any computer that can run dBASE-II. The number of tasks will determine the size of the data base files which, in turn, determines the secondary storage needed. A minimum system would have two floppy-disk drives, each with approximately 500 Kbytes capacity. The multiuser version of ACMS was written to operate on a multiuser Televideo TS 806 with a hard-disk drive of 10 Mbytes.

dBASE-II

The dBASE-II™ assembly language relational data base management system by Ashton-Tate Inc., was used to develop the ACMS. The program is available from several commercial software distributors.

dBASE-II provides a block-structured programming language through which complex input, data processing, and output are possible. The language has control features (IF-ELSE, CASE, DO WHILE) similar to pascal and string manipulation functions (CHR, LEN, VAL, etc.) similar to BASIC that make it very flexible. Data structure and file handling are built-in features of the language that allow for rapid program development. Data structure is limited to data base files that consist of user-defined records containing a maximum of 32 fields. Programs are a sequence of commands written with an editor and stored in an ASCII file. dBASE-II executes these command files interpretively.

Files under the CP/M and MS-DOS operating systems are limited to a maximum of eight characters, then a dot (.), and a three-character extension: FILENAME.EXT. dBASE-II file naming conventions pertinent to ACMS version 84.0 are as follows: data base files have an extension of .DBF; index files of .NDX; command files of .CMD for CP/M-80 or .PRG for CP/M-86 and MS-DOS; and memory variable files of .MEM.

Multiuser Systems

Although the multiuser version of ACMS was developed on the TS 806, it is completely transportable to other multiuser operating systems. The multiuser data base problems of shared file access and file and record lock are handled inside dBASE-II; no operating system-dependent calls are used. ACMS version 84.0 supports only one multiuser data base operation--a two-user Weekly Progress Report (WPR) input. The difficulty in implementing multiuser shared file data base systems independent of the operating system is that, once a user opens a file, most operating systems will make that file read-only for the other users. The file will remain read-only for them until the program returns control to the operating system, at which time the file will be released for write access. If the ACMS.CMD menu system is being used, the operating system will not gain control until this program is terminated. Therefore, data base operations contained in the menu system, except for Weekly Progress Report input, are limited to access by a single user during a session.

The two-user Weekly Progress Report input allows two users, identified as users A and B, to enter reports at the same time. This is implemented using transaction files. The weekly input transaction data base files are of the form W*A.DBF for user A and W*B.DBF for user B (* is the operating system "wildcard" operator). The transaction files are temporary files merged into a single transaction file in the program PS.CMD; this file is then used to update the master. For example, WACTA.DBF and WACTB.DBF are appended into the transaction file WACT.DBF which is used to update the master, ACT.DBF. The main processing program, PS.CMD, clears the weekly transaction files for the next week's data. To understand how these programs work, the command files WPR.CMD, WPR1.CMD, WPR2.CMD, and PS.CMD must be studied (Appendix A). Appendix B gives the structure of the W*.DBF data bases. ACMS can be extended to any number of users by modifying the programs mentioned above and by creating additional transaction files.

The multiuser ACMS system requires that the hard disk have at least two logical drives. As implemented in version 84.0, these drives are A and B, with A as the main working drive. When the user indicates that backup copy of a data base is desired, this backup will be made to the B drive. The exception is BACKUP.CMD, which copies the weekly WPR data onto the floppy drive. On the TS 806, the floppy is C drive.

ACMS version 84.0 is .stributed on two diskettes--one containing the command files and the containing the data base and index files. A directory display of t ıta base diskette should look like that in Figure 1.

```
Name     Ext Bytes   Name    Ext Bytes   Name     Ext  Bytes    Name     Ext Bytes
ACT      DBF    2K ! EQCOST  NDX    2K ! SCHEDULENDX     2K !  WMANDAYSDBF     2K
ACT      NDX    2K ! GROUPS  DBF    2K ! WACT     DBF    2K !  WMANDAYSNDX     2K
BILL     DBF    2K ! GROUPS  NDX    2K ! WACT     NDX    2K !  WMDAYSA DBF     2K
BILL     NDX    2K ! LOCKSA  DBF    2K ! WACTA    DBF    2K !  WMDAYSA NDX     2K
BN       DBF    2K ! LOCKSB  DBF    2K ! WACTA    NDX    2K !  WMDAYSB DBF     2K
BN       NDX    2K ! MANDAYS DBF    2K ! WACTB    DBF    2K !  WMDAYSB NDX     2K
CLUSTERSDBF     2K ! MANDAYS NDX    2K ! WACTB    NDX    2K !  WPR     DBF     2K
CLUSTERSNDX     2K ! MISC    DBF    2K ! WEQ      DBF    2K !  WPR     NDX     2K
DAYS     DBF    2K ! PDF     DBF    2K ! WEQ      NDX    2K !  WPRA    DBF     2K
DAYS     NDX    2K ! PDF     NDX    2K ! WEQA     DBF    2K !  WPRA    NDX     2K
EQ       DBF    2K ! PS      DBF    2K ! WEQA     NDX    2K !  WPRB    DBF     2K
EQ       NDX    2K ! PS      NDX    2K ! WEQB     DBF    2K !  WPRB    NDX     2K
EQCOST   DBF    2K ! SCHEDULEDBF    2K ! WEQB     NDX    2K
51 File(s), occupying 102K of 340K total capacity
13 directory entries and 238K bytes remain on A:
```

Figure 1. Directory display of data base diskette.

Single-User Systems

The single-user version of ACMS eliminates the need for the A and B user WPR transaction files W*A.DBF and W*B.DBF. Only the W*.DBF transaction files are used. The data base distribution diskette directory should appear as shown in Figure 2.

The major consideration in implementing ACMS on a single-user system is that of file referencing by drive. As the system is supplied on the distribution diskette, both command file and data-base file references will be to the logged drive, except for the backup options, which copy to B drive. If ACMS is to be run on a two-floppy system, the user must edit the command files and specify the drives on which the files will be. If ACMS is to be implemented on a system with a hard disk, all command and data-base files can be placed on one hard-disk drive. Backup will be to the B drive unless the command files are edited.

Installation

The first step to take with the two distribution diskettes is to make backup copies. These are the "master working" copies and are to be used only for making other copies. The original masters should be stored in a safe place. A directory display of the command file disk should appear as shown in Figure 3.

If the file referencing must be changed, e.g., command files are to be located on A drive and data base files on B drive, the command files must be edited. To do this, the program listings in Appendix A should be studied. After the files from the distribution diskettes have been copied to the desired drives and DBASE.COM and DBASEOVR.COM have been copied to the drive containing the command files, the program INSTALL.CMD must be run to install printer and console control codes. To run the install program, the user types DBASE INSTALL <CR>. The printer manual should be ready, as the system will prompt for the control codes to change pitch, start emphasized print, and other information. All codes must be entered in ASCII decimal. If the printer does not support a feature requested in the install program, the user

```
Name     Ext Bytes   Name     Ext Bytes   Name     Ext  Bytes    Name     Ext Bytes
ACT      DBF     2K ! DAYS     NDX     2K ! MISC     DBF      2K ! WEQ      DBF      2K
ACT      NDX     2K ! EQ       DBF     2K ! PDF      DBF      2K ! WEQ      NDX      2K
BILL     DBF     2K ! EQ       NDX     2K ! PDF      NDX      2K ! WMANDAYSDBF       2K
BILL     NDX     2K ! EQCOST   DBF     2K ! PS       DBF      2K ! WMANDAYSNDX       2K
BN       DBF     2K ! EQCOST   NDX     2K ! PS       NDX      2K ! WPR      DBF      2K
BN       NDX     2K ! GROUPS   DBF     2K ! SCHEDULEDBF       2K ! WPR      NDX      2K
CLUSTERSDBF      2K ! GROUPS   NDX     2K ! SCHEDULENDX       2K
CLUSTERSNDX      2K ! MANDAYS DBF      2K ! WACT     DBF      2K
DAYS     DBF     2K ! MANDAYS NDX      2K ! WACT     NDX      2K
33 File(s), occupying 66K of 340K total capacity
31 directory entries and 274K bytes remain on D:
```

Figure 2. Data base distribution diskette directory.

```
Name     Ext Bytes   Name     Ext Bytes   Name      Ext   Bytes    Name     Ext Bytes
ACMS     CMD    20K ! EQAUDITCMD    6K ! PDF1      CMD     4K !  SCHEDULECMD   18K
BACKUP   CMD     4K ! EQCOST CMD    8K ! PDF2      CMD     4K !  SPPDF   CMD    4K
BILL     CMD     4K ! EQRPT  CMD    8K ! PDFMNU    CMD    10K !  WCSRPT  CMD    8K
BILLRPT  CMD    10K ! GROUPS CMD   12K ! PDFMNU2   CMD     6K !  WPR     CMD    6K
BNRPT    CMD     8K ! INSTALLCMD   20K ! PPDF      CMD     4K !  WPR1    CMD    4K
CLUSTERSCMD     12K ! IWPR   CMD   24K ! PS        CMD    10K !  WPR2    CMD   14K
COSTRPT  CMD     8K ! LISTPDFCMD    2K ! PSR       CMD     8K
EDEQ     CMD     6K ! MCTRPT CMD   10K ! PSRPT     CMD    12K
EDEQA    CMD     4K ! PDF    CMD    4K ! S3RPT     CMD     8K
33 File(s), occupying 290K of 340K total capacity
27 directory entries and 50K bytes remain on D:
```

Figure 3. Directory display of command file disk.

presses return; the default is a space (ASCII 20) and will not affect the output. The install program stores the control codes in a dBASE-II memory file named INSTALL.MEM. This file will be returned to memory through RESTORE by the menu command file ACMS.CMD, and the other command files will use these memory variables for printer control. The file INSTALL.MEM must be present on the drive containing the command files. The INSTALL.CMD program can be run at any time to redefine the control codes; this program then executes the menu command file ACMS.CMD. After installation, the ACMS system is entered by typing DBASE ACMS <CR>.

Use of Appendices

Users making custom installations of the ACMS version 84.0 should study the appendices thoroughly. Appendix A is a complete listing of all command files in the ACMS system. The single-user commmand files have extensions of .ONE. Appendix B gives the structure of the data base files and the key expressions for the indices. Appendix C shows where data base files are used in the command files.

APPENDIX A:

COMMAND FILE LISTINGS

SINGLE-USER SYSTEM COMMAND FILES

```
*****************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: ACMS.CMD
*
*       DESCRIPTION: This program is an example menu system for the Automated
*                    Construction Management System.  By studying the layout
*                    and the menus, a good understanding of the functional
*                    modules of the ACMS may be obtained.  Users should
*                    customize for their application.
*
*
*       MENUS:                          CALLS:
*
*       Construction Management         WPR.CMD
*
*
*       Project Control                 PDFMNU.CMD, CLUSTERS.CMD, EQCOST.CMD
*                                       IWPR.CMD, BILL.CMD, PS.CMD
*
*       Report Generation               WCSRPT.CMD, S3RPT.CMD,PSR.CMD
*                                       BILLRPT.CMD, BNRPT.CMD, MCTRPT.CMD
*
*       Data Base Backup                BACKUP.CMD
*
*
*****************************************************************************
SET TALK OFF
SET COLON OFF
?
?
?
?
?
?
?
?
?
?
?
?
```

ACMS.CMD

```
?
?
? '       ___________     __________     ___   ___               __________'
? '      /  _____   /    /  ____   /    /   \ /   \             /  ____   /'
? '     /  /    /  /    /  /   /__/    / /\ \/ /\  \           /  /   /__/ '
? '    /  /____/  /    /  /           / /  \  /  \  \         /  /_______  '
? '   /  _____   /    /  /           / /    \/    \  \       /_______   /  '
? '  /  /    /  /    /  /    ___    / /            \  \       ___   /  /   '
? ' /  /    /  /    /  /____/  /   / /              \  \    /  /___/  /    '
? '/__/    /__/    /__________/   /_/                \__\  /_________/     '
?
?
?
?
?
? '                               Version 84.0'
?
?
?
? '                  AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM'
?
? '                         US ARMY CORPS OF ENGINEERS'
?
? '                Construction Engineering Research Laboratory'
?
?
?
STORE 0 TO halt
STORE 50 TO march
DO WHILE halt<march
   STORE halt+1 TO halt
ENDDO
RELEASE ALL LIKE m*
*       * see if ACMS is installed
IF FILE('INSTALL.MEM')
   RESTORE FROM INSTALL
ELSE
*       * error - not installed
ERASE
?
?
?
? '*****THE ACMS INSTALLATION FILE --INSTALL.MEM-- NOT FOUND*****'
?
?
STORE 'Y' TO linst
@ 10,15 SAY  'DO YOU WANT TO RUN THE INSTALL PROGRAM? ' GET linst PICTURE '!'
READ
```

```
IF linst<>'Y'
   ERASE
   QUIT
ELSE
   **********
   DO INSTALL
   **********
ENDIF
ENDIF
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
*************************M-E-N-U****S-T-A-R-T-S****H-E-R-E********************
SET TALK OFF
SET COLON OFF
STORE '1' TO cmd
STORE "OPTION" TO opt
DO WHILE cmd<> 'Q0'
   ERASE
   STORE '1' TO cmd
   STORE "CONSTRUCTION MANAGEMENT" TO menu
   @  1, 9 SAY "**********************************************************"
   @  2, 9 SAY "*             "+orgname
   @  2,66 SAY "*"
   @  3, 9 SAY "*        AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM        *"
   @  4, 9 SAY "*            < < <"
   @  4,28 GET menu
   CLEAR GETS
   @  4,51 SAY " > > >         *"
   @  5, 9 SAY "**********************************************************"
   @  6, 9 SAY "*                                                        *"
   @  7, 9 SAY "*    <1>  PROJECT CONTROL                                *"
   @  8, 9 SAY "*                                                        *"
   @  9, 9 SAY "*    <2>  REPORT GENERATION                              *"
   @ 10, 9 SAY "*                                                        *"
   @ 11, 9 SAY "*    <3>  INPUT WEEKLY PROGRESS REPORTS                  *"
   @ 12, 9 SAY "*                                                        *"
   @ 13, 9 SAY "*    <4>  DATA BASE BACKUP                               *"
   @ 14, 9 SAY "*                                                        *"
   @ 15, 9 SAY "*    <0>  QUIT                                           *"
   @ 16, 9 SAY "*                                                        *"
   @ 17, 9 SAY "*                                                        *"
   @ 18, 9 SAY "*"
   @ 18,31 GET opt
   CLEAR GETS
   @ 18,39 SAY "<"
   @ 18,40 GET cmd PICTURE '9'
   @ 18,41 SAY ">                        *"
   @ 19, 9 SAY "**********************************************************"
   READ
   CLEAR GETS

   DO CASE


      CASE cmd= '0'

         ERASE
         QUIT


      CASE cmd= '1'

         DO WHILE cmd<> 'Q2'

            STORE '1' TO cmd
            STORE "PROJECT CONTROL" TO menu
            ERASE
@  1, 9 SAY "**********************************************************"
@  2, 9 SAY "*             "+orgname
@  2,66 SAY "*"
@  3, 9 SAY "*        AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM          *"
@  4, 9 SAY "*              < < < "
@  4,30 GET menu
CLEAR GETS
@  4,46 SAY "> > >             *"
@  5, 9 SAY "**********************************************************"
@  6, 9 SAY "*                                                        *"
@  7, 9 SAY "*    <1>  PROJECT DATA BASE                              *"
@  8, 9 SAY "*                                                        *"
@  9, 9 SAY "*    <2>  REPORTING GROUPS ROSTER                        *"
@ 10, 9 SAY "*                                                        *"
@ 11, 9 SAY "*    <3>  EQUIPMENT DATA BASE                            *"
@ 12, 9 SAY "*                                                        *"
@ 13, 9 SAY "*    <4>  INITIAL WEEKLY PROGRESS REPORT                 *"
@ 14, 9 SAY "*                                                         *"
@ 15, 9 SAY "*    <5>  CUSTOMER BILLING DATA BASE                       *"
@ 16, 9 SAY "*                                                         *"
@ 17, 9 SAY "*    <6>  -- PROCESS WPR'S --                              *"
@ 18, 9 SAY "*                                                         *"
@ 19, 9 SAY "*    <0>  QUIT                                             *"
@ 20, 9 SAY "*                                                         *"
@ 21, 9 SAY "*"
@ 21,31 GET opt
CLEAR GETS
@ 21,39 SAY "<"
@ 21,40 GET cmd PICTURE '9'
@ 21,41 SAY ">                         *"
@ 22, 9 SAY "***********************************************************"
            READ
            CLEAR GETS

            DO CASE

               CASE cmd= '1'

                  *********
                  DO pdfmnu
                  *********
                  *     * make sure prim and second are clear
                  SELECT PRIMARY
                  USE
                  SELECT SECONDARY
                  USE

               CASE cmd= '2'

                  ***********
                  DO clusters
                  ***********
                  *     * make sure prim and second are clear
                  SELECT PRIMARY
                  USE
                  SELECT SECONDARY
                  USE

               CASE cmd= '3'

                  *********
                  DO eqcost
                  *********
                  *     * make sure prim and second are clear
                  SELECT PRIMARY
                  USE
                  SELECT SECONDARY
                  USE
               CASE cmd= '4'

                  *******
                  DO iwpr
                  *******
                  *     * make sure prim and second are clear
                  SELECT PRIMARY
                  USE
                  SELECT SECONDARY
                  USE

               CASE cmd= '5'

                  *******
                  DO bill
                  *******
                  *     * make sure prim and second are clear
                  SELECT PRIMARY
                  USE
                  SELECT SECONDARY
                  USE

               CASE cmd= '6'

                  *****
                  DO ps
                  *****
                  *     * make sure prim and second are clear
                  SELECT PRIMARY
                  USE
                  SELECT SECONDARY
                  USE

               CASE cmd= '0'

                  STORE 'Q2' TO cmd
                  LOOP

            ENDCASE

         ENDDO WHILE cmd<> 'Q2'


      CASE cmd= '2'

         DO WHILE cmd<> 'Q3'

            STORE '1' TO cmd
            STORE "REPORT GENERATION" TO menu
            ERASE
@  1, 9 SAY "**********************************************************"
@  2, 9 SAY "*             "+orgname
@  2,66 SAY "*"
@  3, 9 SAY "*        AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM         *"
@  4, 9 SAY "*             < < < "
@  4,29 GET menu
CLEAR GETS
@  4,47 SAY "> > >              *"
@  5, 9 SAY "**********************************************************"
@  6, 9 SAY "*                                                        *"
@  7, 9 SAY "*    <1>  WEEKLY CONSTRUCTION STATUS REPORT              *"
@  8, 9 SAY "*                                                        *"
@  9, 9 SAY "*    <2>  S-3 WEEKLY REPORT                              *"
@ 10, 9 SAY "*                                                        *"
@ 11, 9 SAY "*    <3>  PROJECT STATUS REPORT                          *"
@ 12, 9 SAY "*                                                        *"
@ 13, 9 SAY "*    <4>  CUSTOMER BILLING REPORT                        *"
@ 14, 9 SAY "*                                                        *"
@ 15, 9 SAY "*    <5>  BATTALION COST REPORT                          *"
@ 16, 9 SAY "*                                                        *"
@ 17, 9 SAY "*    <6>  MONTHLY TROOP CONSTRUCTION REPORT              *"
@ 18, 9 SAY "*                                                        *"
@ 19, 9 SAY "*    <0>  QUIT                                           *"
@ 20, 9 SAY "*                                                        *"
@ 21, 9 SAY "*"
@ 21,31 GET opt
CLEAR GETS
@ 21,39 SAY "<"
@ 21,40 GET cmd PICTURE '9'
@ 21,41 SAY ">                        *"
@ 22, 9 SAY "**********************************************************"
            READ
            CLEAR GETS

            DO CASE

               CASE cmd= '1'

                  *********
                  DO wcsrpt
                  *********
                  *     * make sure prim and second are clear
                  SELECT PRIMARY
                  USE
                  SELECT SECONDARY
                  USE

               CASE cmd= '2'

                  ********
                  DO s3rpt
                  ********
                  *     * make sure prim and second are clear
                  SELECT PRIMARY
```

```
   USE
   SELECT SECONDARY
   USE

CASE cmd= '3'

   ******
   DO psr
   ******
   *     * make sure prim and second are clear
   SELECT PRIMARY
   USE
   SELECT SECONDARY
   USE

CASE cmd= '4'

   **********
   DO billrpt
   **********
   *     * make sure prim and second are clear
   SELECT PRIMARY
   USE
   SELECT SECONDARY
   USE

CASE cmd= '5'

   ********
   DO bnrpt
   ********
   *     * make sure prim and second are clear
   SELECT PRIMARY
   USE
   SELECT SECONDARY
   USE

CASE cmd= '6'

   *********
   DO mctrpt
   *********
   *     * make sure prim and second are clear
   SELECT PRIMARY
   USE
   SELECT SECONDARY
   USE

CASE cmd= '0'

   STORE 'Q3' TO cmd
   LOOP
```

```
            ENDCASE

         ENDDO WHILE cmd<> 'Q3'


      CASE cmd= '3'

         ******
         DO wpr
         ******
         *      * make sure prim and second are clear
         SELECT PRIMARY
         USE
         SELECT SECONDARY
         USE

      CASE cmd= '4'



                  DO WHILE cmd<> 'Q5'
                     ERASE
                     STORE '1' TO cmd
                     STORE "DATA BASE BACKUP" TO menu
@  1, 9 SAY "**********************************************************"
@  2, 9 SAY "*             "+orgname
@  2,66 SAY "*"
@  3, 9 SAY "*        AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM        *"
@  4, 9 SAY "*              < < < "
@  4,30 GET menu
CLEAR GETS
@  4,47 SAY "> > >              *"
@  5, 9 SAY "**********************************************************"
@  6, 9 SAY "*                                                        *"
@  7, 9 SAY "*    <1>  AFTER PROCESSING DATA BASE BACKUP (TO A:)      *"
@  8, 9 SAY "*                                                        *"
@  9, 9 SAY "*    <2>  BACKUP EQUIPMENT COST DATA BASE (TO B:)        *"
@ 10, 9 SAY "*                                                        *"
@ 11, 9 SAY "*    <3>  BACKUP IWPR DATA BASE (TO B:)                  *"
@ 12, 9 SAY "*                                                        *"
@ 13, 9 SAY "*    <4>  BACKUP PROJECT DATA BASE (TO B:)               *"
@ 14, 9 SAY "*                                                        *"
@ 15, 9 SAY "*    <0>  QUIT                                           *"
@ 16, 9 SAY "*                                                        *"
@ 17, 9 SAY "*"
@ 17,31 GET opt
CLEAR GETS
@ 17,39 SAY "<"
@ 17,40 GET cmd PICTURE '9'
@ 17,41 SAY ">                        *"
@ 18, 9 SAY "**********************************************************"
```

```
READ
CLEAR GETS

DO CASE

   CASE cmd= '1'

      *********
      DO backup
      *********
      *    * make sure prim and second are clear
      SELECT PRIMARY
      USE
      SELECT SECONDARY
      USE

   CASE cmd= '2'

      SELECT PRIMARY
      USE
      USE eqcost
      COPY STRUCTURE TO b:eqcost
      USE b:eqost
      INDEX ON STR(eno,4,0)+STR(atchno,2,0) TO b:eqcost
      SET INDEX TO b:eqcost
      APPEND FROM a:eqcost
      USE

   CASE cmd= '3'

      SELECT PRIMARY
      USE
      USE a:act
      COPY STRUCTURE TO b:act
      USE b:act
      INDEX ON sno+str(i,4,0)+str(j,4,0) TO b:act
      SET INDEX TO b:act
      APPEND FROM a:act
      USE

   CASE cmd= '4'

      SELECT PRIMARY
      USE
      USE pdf
      COPY STRUCTURE TO b:pdf
      USE b:pdf
      INDEX ON sno TO b:pdf
      SET INDEX TO b:pdf
      APPEND FROM a:pdf
      USE
      USE ps
```

```
                           COPY STRUCTURE TO b:ps
                           USE b:ps
                           INDEX ON sno TO b:ps
                           SET INDEX TO b:ps
                           APPEND FROM a:ps
                           USE

                       CASE cmd= '0'

                           STORE 'Q5' TO cmd
                           LOOP

                     ENDCASE

                  ENDDO WHILE cmd<> 'Q5'


   ENDCASE

ENDDO WHILE cmd<> 'Q0'
*
*       * END ACMS.CMD
*
```

```
********************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                 USA-CERL
*
*       PROGRAM: BACKUP.CMD
*
*       DESCRIPTION: Data base backup program to clear processed data from
*                    weekly temporary data bases.  All weekly input data base
*                    file names begin with "W".  Backs off data contained in
*                    WPR.DBF, WMANDAYS.DBF, WACT.DBF, and WEQ.DBF.  Copies all
*                    data entered on or after the last processing date to files
*                    of the type W???mmdd.DBF, where mmdd is the last
*                    processing date.  The program copies these files to the
*                    floppy drive ( C: ) and a COPY of the project status
*                    data base PS.CMD.
*
*
*       CALLS: none
*
********************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
@ 1,0
? '                      WEEKLY DATA BASE BACKUP'
CLEAR GETS
*       * get date of last processing
USE misc
STORE date TO mdate
USE
*       * all created .dbf files have month and day in file name
*       * example: WPR0708.DBF <= WPR&mmdd.DBF
STORE $(mdate,1,4) TO mmdd

@ 5,0
? '                      Backing off CERL data '
USE wpr INDEX wpr
COPY TO wpr&mmdd FOR week<=mdate
COPY TO tt FOR week>mdate
USE tt
COPY TO wpr
USE wpr
INDEX ON sno+week TO wpr

? '                      Backing off old mandays data'
USE wmandays INDEX wmandays
COPY TO wman&mmdd FOR week<=mdate
COPY TO tt FOR week>mdate
USE tt
```

```
COPY TO wmandays
USE wmandays
INDEX ON sno+week TO wmandays

? '                       Backing off old CPM data'
USE wact INDEX wact
COPY TO wact&mmdd FOR week<=mdate
COPY TO tt FOR week>mdate
USE tt
COPY TO wact
USE wact
INDEX ON sno+week+str(i,4,0)+str(j,4,0) TO wact

? '                       Backing off old equipment data'
USE weq INDEX weq
COPY TO weq&mmdd FOR week<=mdate
COPY TO tt FOR week>mdate
USE eq index eq
append from weq&mmdd
USE tt
COPY TO weq
USE weq
INDEX ON sno+week TO weq
USE
DELETE FILE tt
ERASE
@ 1,0
?
? "                Insert Diskette in Drive to make CERL Data Disk"
?
? "                       Press RETURN when ready to copy"
SET CONSOLE OFF
WAIT
SET CONSOLE ON
?
? "                              WAIT...COPYING"
*       * floppy drive is C: on Televideo TS 806
SET DEFA TO C:
RESET C:
SET DEFA TO A:
USE wpr&mmdd
COPY to c:wpr&mmdd
USE wman&mmdd
COPY to c:wman&mmdd
USE weq&mmdd
COPY to c:weq&mmdd
USE  wact&mmdd
COPY to c:wact&mmdd
USE
*       * if file has been copied to floppy, delete from hard disk (A:)
IF FILE ('c:wpr&mmdd')
DELETE FILE a:wpr&mmdd
```

```
ENDIF
IF FILE ('c:wman&mmdd')
DELETE FILE a:wman&mmdd
ENDIF
IF FILE ('c:weq&mmdd')
DELETE FILE a:weq&mmdd
ENDIF
IF FILE ('c:wact&mmdd')
DELETE FILE a:wact&mmdd
ENDIF
USE
*       * get copy of project status data base for analysis
USE PS
COPY TO c:ps&mmdd
USE
RETURN
*
*       * END BACKUP.CMD
*
```

```
******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: BILL.CMD
*
*       DESCRIPTION: This program allows the user to set up customer billing
*                    system based on construction Project Number.  Charges
*                    against all projects having the same project number
*                    in the project data base (PDF.DBF) will be accumulated
*                    in BILL.DBF, one record for each Project Number.
*                    Also stores header information for the bill.
*
*       CALLS: None
*
******************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
SET DELETE ON
SELECT PRIMARY
USE bill INDEX bill

SELECT SECOND
USE pdf

STORE "XXXXXXXXXXXXXX" TO mpno
STORE "Y" TO mmore

DO WHILE !(mmore)= "Y"
   ERASE
   @ 1,22 SAY "CUSTOMER BILLING"
   @ 3, 5 SAY "PN  Number : " GET mpno
   *    * get directive number form user
   READ
   CLEAR GETS
   SELECT SECOND
   LOCATE FOR pno=mpno

   IF pno<>mpno
      @ 4,14 SAY "Unknown PNO Number.  Try again? (y/n) " GET mmore
      READ
      IF !(mmore)<>'Y'
        STORE 'N' TO  mmore
        LOOP
        CLEAR GETS
      ENDIF
   ELSE
      SELECT PRIMARY
      FIND "&mpno"

      IF #=0
         *     * not in data base so add it
         APPEND BLANK
         REPLACE pno WITH mpno, FROM WITH "XXXXXX" , billno WITH 1
      ENDIF #=0
         *     * display header information
         @ 5, 0 SAY "BILLING NUMBER:"
         @ 5,16 SAY billno
         @ 5,39 SAY "From: " +FROM
         @ 7, 0 SAY "DA FORM 2544:"
         @ 7,14 GET da
         @ 7,39 SAY "DATE:"
         @ 7,45 GET DATE
         @ 9, 0 SAY "TITLE:"
         @ 9, 7 GET title
         @ 9,39 SAY "UNIT:"
         @ 9,45 GET unit
         @ 11,0 SAY "Schedules Included: "
         ?? "                 <Type CTR-Q to abort>"
         SELECT SECOND
         STORE 12 TO row
         STORE 0 TO col
         *      up to 80 schedules can be displayed on the crt without
         *      destroying the screen format.

         DO WHILE .NOT.EOF
            @ row,col SAY sno
            IF col=80
               STORE 0 TO col
               STORE row+1 TO row
            ELSE
               STORE col+10 TO col
            ENDIF col=80
            CONTINUE
         ENDDO WHILE .NOT.EOF

         *      * display edit control keys for user
@ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field;
    ;  CTR-D: next character ;"
@ 23, 5 SAY "CTR-W: next screen ;  CTR-S: previous character ;"

         SELECT PRIMARY
         READ
         CLEAR GETS
         @ 20,0
         @ 21,0
         @ 20,7 SAY "continue? (y/n) " GET mmore
         READ
         CLEAR GETS

   ENDIF pno<>mpno

ENDDO !(mmore)= "Y"

ERASE
RETURN
*
*       * END BILL.CMD
*
```

```
******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                   USA-CERL
*
*       PROGRAM: BILLRPT.CMD
*
*       DESCRIPTION: Prints a bill for each Project Number in BILL.DBF.
*                    Billing number is incremented each time the report is
*                    generated.  Designed as a weekly report.  Stores cumulatives
*                    in BILL.DBF.
*
*
*       CALLS: None
*
******************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
SET DELETE ON
*        * get last processing date
USE misc
GO TOP
STORE date TO mdate

@ 1,0 SAY '                  PROCESS WEEKLY PROGRESS REPORTS'
@ 4,0 SAY '                         Customer Billing'
@ 5,0 SAY '                     For Period Ending:' GET mdate
CLEAR GETS
@ 8,29 SAY 'PROCESSING'
*        * string memory variables of the type list# are used to store
*        * schedule numbers of tasks included in the bill
*        * they are written to disk files list#.mem
*        * erase any old list&no..mem files
STORE '1' TO no
DO WHILE &no < 10
   IF FILE( 'list&no..mem' )
      DELETE FILE LIST&no..mem
   ENDIF FILE( 'list&no..mem' )
   STORE str(&no+1,1,0) TO no
ENDDO WHILE &no < 10

SELECT SECOND
USE pdf INDEX pdf

SELECT PRIMARY
USE bill INDEX bill
GOTO TOP
```

```
DO WHILE .NOT.EOF
   STORE pno TO mpno
   *        * mac?: variables used to sum 'allocated funds' ( ac? in ps.dbf)
   STORE 0 TO mac1
   STORE 0 TO mac2
   STORE 0 TO mac3
   STORE 0 TO mac4
   STORE 0 TO mac5
   *        * mec?: variables used to sum 'expanded funds' ( ec? in ps.dbf)
   STORE 0 TO mec1
   STORE 0 TO mec2
   STORE 0 TO mec3
   STORE 0 TO mec4
   STORE 0 TO mec5
   @ 12,25 SAY 'Project Number:' GET mpno
   CLEAR GETS
   SELECT PRIMARY
   USE ps INDEX ps
   SELECT SECOND
   LOCATE FOR pno=mpno
   STORE '1' TO no
   *       * snolist is a memory variable string to store
   *       * schedule numbers used
   *       * maximum string lenght is 255 characters
   STORE "    " TO snolist&no
   STORE customer TO mcustomer

   DO WHILE .NOT.EOF

      STORE sno TO msno
      IF len(snolist&no)<210
         STORE snolist&no+msno+ "    " TO snolist&no
      ELSE
         SAVE TO LIST&no ALL LIKE snolist&no
                 RELEASE snolist&no
         STORE str(&no+1,1,0) TO no
         STORE "    " +msno+ "    " TO snolist&no
      ENDIF len(snolist&no)<240


      SELECT PRIMARY
      FIND "&msno"

      STORE mac1+ac1 TO mac1
      STORE mac2+ac2 TO mac2
      STORE mac3+ac3 TO mac3
      STORE mac4+ac4 TO mac4
      STORE mac5+ac5 TO mac5

      STORE mec1+ec1 TO mec1
      STORE mec2+ec2 TO mec2
      STORE mec3+ec3 TO mec3
      STORE mec4+ec4 TO mec4
      STORE mec5+ec5 TO mec5

      SELECT SECOND
      CONTINUE

   ENDDO WHILE .NOT.EOF

   SELECT PRIMARY
   USE bill INDEX bill
   FIND "&mpno"
   STORE mac1+mac2+mac3+mac4+mac5 TO mac6
   STORE mec1+mec2+mec3+mec4+mec5 TO mec6
   STORE ustdy+usequip+csgtdy+csgequip+csglbr TO TOTAL

   IF len(snolist1)=4
      *         * no schedules with dno=mpno
      SET FORMAT TO PRINT
      @ 15,3 SAY "Warning: no schedules found for directive no.: &mpno"
      EJECT
      SET FORMAT TO SCREEN
   ELSE

*-----SEND REPORT TO PRINTER--------------------------------------------------
      SET FORMAT TO PRINT
      SET MARGIN TO 0

      *         * bill_this_period = cost_todate - previous_bill
      *         * b?: billings this time
      STORE mec1-ustdy TO b1
      STORE mec2-usequip TO b2
      STORE mec3-csgtdy TO b3
      STORE mec4-csgequip TO b4
      STORE mec5-csglbr TO b5
      *         * ba?: balance this time
      STORE mac1-mec1 TO ba1
      STORE mac2-mec2 TO ba2
      STORE mac3-mec3 TO ba3
      STORE mac4-mec4 TO ba4
      STORE mac5-mec5 TO ba5
      STORE mec6-TOTAL TO b6
      STORE mac6-mec6 TO ba6
      *
@ 4,22 SAY "CUSTOMER BILLING"
@ 8, 0 SAY "PN    NUMBER:"
@ 8,14 SAY mpno
@ 8,39 SAY "CUSTOMER: &mcustomer"
@ 10, 0 SAY "BILLING NUMBER:"
@ 10,16 SAY billno
@ 10,39 SAY "From:"
@ 10,45 SAY FROM
@ 10,52 SAY "To:"
@ 10,56 SAY mdate
@ 12, 0 SAY "DA FORM 2544:"
@ 12,14 SAY da
@ 12,39 SAY "DATE:"
@ 12,45 SAY DATE
@ 14, 0 SAY "TITLE:"
@ 14, 7 SAY title
@ 14,39 SAY "UNIT:"
@ 14,45 SAY unit
@ 17, 0 SAY "+-----------+-----------+-----------+------------+-----------+---
--------+"
@ 18, 0 SAY "|           |           | PROJECT   | PREVIOUS   | BILLING
|           |"
@ 19, 0 SAY "| TYPE OF   | ALLOCATED | COST      | CUMULATIVE | THIS      |
BALANCE    |"
@ 20, 0 SAY "| FUNDS     | FUNDS     | TO DATE   | BILLING    | PERIOD
|           |"
@ 21, 0 SAY "+-----------+-----------+-----------+------------+-----------+---
--------+"
@ 22, 0 SAY "| US TDY    |"
@ 22,13 SAY mac1
@ 22,24 SAY "|"
@ 22,25 SAY mec1
@ 22,36 SAY "|"
@ 22,37 SAY ustdy
@ 22,49 SAY "|"
@ 22,50 SAY b1
@ 22,61 SAY "|"
@ 22,62 SAY ba1
@ 22,73 SAY "|"
@ 23, 0 SAY "+-----------+-----------+-----------+------------+-----------+---
--------+"
@ 24, 0 SAY "| US EQUIP  |"
@ 24,13 SAY mac2
@ 24,24 SAY "|"
@ 24,25 SAY mec2
@ 24,36 SAY "|"
@ 24,37 SAY usequip
@ 24,49 SAY "|"
@ 24,50 SAY b2
@ 24,61 SAY "|"
@ 24,62 SAY ba2
@ 24,73 SAY "|"
@ 25, 0 SAY "+-----------+-----------+-----------+------------+-----------+---
--------+"
@ 26, 0 SAY "| CSG TDY   |"
@ 26,13 SAY mac3
@ 26,24 SAY "|"
@ 26,25 SAY mec3
@ 26,36 SAY "|"
@ 26,37 SAY csgtdy
@ 26,49 SAY "|"
@ 26,50 SAY b3
@ 26,61 SAY "|"
@ 26,62 SAY ba3
@ 26,73 SAY "|"
@ 27, 0 SAY "+-----------+-----------+-----------+------------+-----------+---
--------+"
@ 28, 0 SAY "| CSG EQUIP |"
@ 28,13 SAY mac4
@ 28,24 SAY "|"
@ 28,25 SAY mec4
@ 28,36 SAY "|"
@ 28,37 SAY csgequip
@ 28,49 SAY "|"
@ 28,50 SAY b4
@ 28,61 SAY "|"
@ 28,62 SAY ba4
@ 28,73 SAY "|"
@ 29, 0 SAY "+-----------+-----------+-----------+------------+-----------+---
--------+"
@ 30, 0 SAY "| CSG LABOR |"
@ 30,13 SAY mac5
@ 30,24 SAY "|"
@ 30,25 SAY mec5
@ 30,36 SAY "|"
@ 30,37 SAY csglbr
@ 30,49 SAY "|"
@ 30,50 SAY b5
@ 30,61 SAY "|"
@ 30,62 SAY ba5
@ 30,73 SAY "|"
@ 31, 0 SAY "+-----------+-----------+-----------+------------+-----------+---
--------+"
@ 32, 0 SAY "| TOTAL     |"
@ 32,13 SAY mac6
@ 32,24 SAY "|"
@ 32,25 SAY mec6
@ 32,36 SAY "|"
@ 32,37 SAY TOTAL
@ 32,49 SAY "|"
@ 32,50 SAY b6
@ 32,61 SAY "|"
@ 32,62 SAY ba6
@ 32,73 SAY "|"
@ 33, 0 SAY "+-----------+-----------+-----------+------------+-----------+---
--------+"

@ 40,0 SAY "Tasks Included:"
   STORE '1' TO mi
   STORE 2 TO row
   DO WHILE &mi<=&no
      IF FILE( 'list&mi..mem' )
         RESTORE FROM LIST&mi ADDITIVE
         DELETE FILE LIST&mi..mem
      ENDIF FILE( 'list&mi..mem' )
      *         * print out schedule numbers included in bill
      @ 40+row,0 SAY $(snolist&mi,1,70)
      STORE row+1 TO row
      IF len(snolist&mi)>70
         @ 40+row,0 SAY $(snolist&mi,71,70)
         STORE row+1 TO row
         IF len(snolist&mi)>140
            @ 40+row,0 SAY $(snolist&mi,141,70)
            STORE row+1 TO row
         ENDIF len(snolist&mi)>200
      ENDIF len(snolist&mi)>50
      RELEASE ALL LIKE snolist&mi
      STORE str(&mi+1,1,0) TO mi
   ENDDO WHILE &mi<=&no


  EJECT
  SET FORMAT TO SCREEN

  *     * increment bill number
  REPLACE billno WITH billno+1
  *     * store current date for next bill
  REPLACE from WITH mdate
  *     * store cumulative charges
  REPLACE ustdy WITH mec1,usequip WITH mec2,csgtdy WITH mec3
  REPLACE csgequip WITH mec4, csglbr WITH mec5
  SKIP +1

ENDDO WHILE .NOT.EOF

ERASE
RETURN
*
*       * END BILLPRT.CMD
*
```

```
******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                   USA-CERL
*
*       PROGRAM: BNRPT.CMD
*
*       DESCRIPTION: Generates a battalion cost report for each battalion.
*                    One record per battalion is generated by PDF.CMD as
*                    tasks are entered.  This program is similar to
*                    BILLRPT.CMD.  The battalion record is updated each
*                    time the report is printed.
*
*       CALLS: none
*
******************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
SET DELETE ON
*       * get printer/console control codes
RESTORE FROM install ADDITIVE
*       * get date of last processing
USE misc
STORE date TO mdate
*       * routine to convert mmddyy to military date ddmmmyy
STORE 'JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC' TO year
STORE $(mdate,1,2) TO mnth
STORE $(mdate,3,2)+ ' ' +$(year,(val(mnth)*3-2),3)+ ' ' +$(mdate,5,2) TO mil
@ 1,0 SAY '                        BATTALION COST REPORT'
@ 3,0 SAY '                      For Period Ending: ' GET mil
CLEAR GETS
@ 6,29 SAY rv+'PROCESSING'+rvoff
*       * release any old list#.mem files
STORE '1' TO no
DO WHILE &no < 10
   IF FILE( 'list&no..mem' )
      DELETE FILE LIST&no..mem
   ENDIF FILE( 'list&no..mem' )
   STORE str(&no+1,1,0) TO no
ENDDO WHILE &no < 10
USE
SELECT SECOND
USE pdf
INDEX ON bn TO pdf.bn
SET INDEX TO pdf.bn
GO TOP
STORE bn TO mbn
STORE sno TO msno
```

```
@ 12,25 SAY '   Battalion : ' GET mbn
CLEAR GETS
STORE t TO c
STORE t TO b

DO WHILE b
   STORE '1' TO no
   STORE "    " TO snolist&no

   STORE 0 TO mec1
   STORE 0 TO mec2
   STORE 0 TO mec3
   STORE 0 TO mec4
   STORE 0 TO mec5
   STORE 0 TO mac1
   STORE 0 TO mac2
   STORE 0 TO mac3
   STORE 0 TO mac4
   STORE 0 TO mac5

   SELECT PRIMARY
   USE ps INDEX ps
   DO WHILE c
      SELECT PRIMARY
      FIND '&msno'
      STORE ec1+mec1 TO mec1
      STORE ec2+mec2 TO mec2
      STORE ec3+mec3 TO mec3
      STORE ec4+mec4 TO mec4
      STORE ec5+mec5 TO mec5
      STORE ac1+mac1 TO mac1
      STORE ac2+mac2 TO mac2
      STORE ac3+mac3 TO mac3
      STORE ac4+mac4 TO mac4
      STORE ac5+mac5 TO mac5

      IF len(snolist&no)<210
         STORE snolist&no+msno+ "    " TO snolist&no
      ELSE
         SAVE TO LIST&no ALL LIKE snolist&no
         RELEASE snolist&no
         STORE str(&no+1,1,0) TO no
         STORE "    " +msno+ "    " TO snolist&no
      ENDIF len(snolist&no)<240

      SELECT SECOND
      SKIP +1

      IF s.bn<>mbn
         STORE bn TO mbn2
         STORE # TO r
         STORE f TO c
      ELSE
         STORE sno TO msno
      ENDIF s.bn<>mbn

      IF EOF
         STORE f TO c
         STORE f TO b
      ENDIF

   ENDDO WHILE c

   SELECT PRIMARY
   USE bn INDEX bn
   FIND "&mbn"
   STORE mac1+mac2+mac3+mac4+mac5 TO mac6
   STORE mec1+mec2+mec3+mec4+mec5 TO mec6
   STORE ustdy+usequip+csgtdy+csgequip+csglbr TO TOTAL

   *----------------------------send report to printer-------------------------
   SET FORMAT TO PRINT
   SET MARGIN TO 0

   *        * bill_this_period = cost_todate - previous_bill
   *        * the previous bills have always been balanced.
   *        * b?: billings this atime
   STORE mec1-ustdy TO b1
   STORE mec2-usequip TO b2
   STORE mec3-csgtdy TO b3
   STORE mec4-csgequip TO b4
   STORE mec5-csglbr TO b5
   *        * ba?: balance this time
   STORE mac1-mec1 TO ba1
   STORE mac2-mec2 TO ba2
   STORE mac3-mec3 TO ba3
   STORE mac4-mec4 TO ba4
   STORE mac5-mec5 TO ba5
   STORE mec6-TOTAL TO b6
   STORE mac6-mec6 TO ba6
   STORE FROM TO mfrom
   *        * routine to convert mmddyy to military date ddmmmyy
   STORE $(mfrom,1,2) TO mnth
   STORE $(mfrom,3,2)+ ' ' +$(year,(val(mnth)*3-2),3)+ ' ' +$(mfrom,5,2) TO
fromil
   *
   @ 4,22 SAY "BATTALION COSTS"
   @ 8, 0 SAY "BATTALION :"
   @ 8,12 SAY mbn
   @ 8,39 3AY "From:"
   @ 8,45 SAY fromil
   @ 10, 0 SAY "REPORT NUMBER:"
   @ 10,14 SAY billno
   @ 10,39 SAY "To:"
   @ 10,45 SAY mil
   @ 17, 0 SAY "+-----------+-----------+-----------+------------+-----------
+-----------+"
   @ 18, 0 SAY "|           |           |           | PREVIOUS   |   COST
|           |"
   @ 19, 0 SAY "|  TYPE OF  | ALLOCATED |   COST    | CUMULATIVE |   THIS    |
BALANCE    |"
   @ 20, 0 SAY "|   FUNDS   |   FUNDS   | TO DATE   | COSTS      |   PERIOD
|           |"
   @ 21, 0 SAY "+-----------+-----------+-----------+------------+-----------
+-----------+"
   @ 22, 0 SAY "| US TDY    |"
   @ 22,13 SAY mac1
   @ 22,24 SAY "|"
   @ 22,25 SAY mec1
   @ 22,36 SAY "|"
   @ 22,37 SAY ustdy
   @ 22,49 SAY "|"
   @ 22,50 SAY b1
   @ 22,61 SAY "|"
   @ 22,62 SAY ba1
   @ 22,73 SAY "|"
   @ 23, 0 SAY "+-----------+-----------+-----------+------------+-----------
+-----------+"
   @ 24, 0 SAY "| US EQUIP  |"
   @ 24,13 SAY mac2
   @ 24,24 SAY "|"
   @ 24,25 SAY mec2
   @ 24,36 SAY "|"
   @ 24,37 SAY usequip
   @ 24,49 SAY "|"
   @ 24,50 SAY b2
   @ 24,61 SAY "|"
   @ 24,62 SAY ba2
   @ 24,73 SAY "|"
   @ 25, 0 SAY "+-----------+-----------+-----------+------------+-----------
+-----------+"
   @ 26, 0 SAY "| CSG TDY   |"
   @ 26,13 SAY mac3
   @ 26,24 SAY "|"
   @ 26,25 SAY mec3
   @ 26,36 SAY "|"
   @ 26,37 SAY csgtdy
   @ 26,49 SAY "|"
   @ 26,50 SAY b3
   @ 26,61 SAY "|"
   @ 26,62 SAY ba3
   @ 26,73 SAY "|"
   @ 27, 0 SAY "+-----------+-----------+-----------+------------+-----------
+-----------+"
   @ 28, 0 SAY "| CSG EQUIP |"
   @ 28,13 SAY mac4
   @ 28,24 SAY "|"
   @ 28,25 SAY mec4
   @ 28,36 SAY "|"
   @ 28,37 SAY csgequip
   @ 28,49 SAY "|"
   @ 28,50 SAY b4
   @ 28,61 SAY "|"
   @ 28,62 SAY ba4
   @ 28,73 SAY "|"
   @ 29, 0 SAY "+------------+-----------+-----------+------------+-----------
+-----------+"
   @ 30, 0 SAY "| CSG LABOR |"
   @ 30,13 SAY mac5
   @ 30,24 SAY "|"
   @ 30,25 SAY mec5
   @ 30,36 SAY "|"
   @ 30,37 SAY csglbr
   @ 30,49 SAY "|"
   @ 30,50 SAY b5
   @ 30,61 SAY "|"
   @ 30,62 SAY ba5
   @ 30,73 SAY "|"
   @ 31, 0 SAY "+------------+-----------+-----------+------------+-----------
+-----------+"
   @ 32, 0 SAY "| TOTAL     |"
   @ 32,13 SAY mac6
   @ 32,24 SAY "|"
   @ 32,25 SAY mec6
   @ 32,36 SAY "|"
   @ 32,37 SAY TOTAL
   @ 32,49 SAY "|"
   @ 32,50 SAY b6
   @ 32,61 SAY "|"
   @ 32,62 SAY ba6
   @ 32,73 SAY "|"
   @ 33, 0 SAY "+------------+-----------+-----------+------------+-----------
+-----------+"
   @ 35,0 SAY "Tasks Included:"
   *   * print tasks included in this report
   STORE '1' TO mi
   STORE 0 TO row
   DO WHILE &mi<=&no
      IF FILE( 'list&mi..mem' )
         RESTORE FROM LIST&mi additive
         DELETE FILE LIST&mi..mem
      ENDIF FILE( 'list&mi..mem' )
      * print out schedule numbers included in bill
      @ 40+row,0 SAY $(snolist&mi,1,70)
      STORE row+1 TO row
      IF len(snolist&mi)>70
         @ 40+row,0 SAY $(snolist&mi,71,70)
         STORE row+1 TO row
         IF len(snolist&mi)>140
            @ 40+row,0 SAY $(snolist&mi,141,70)
            STORE row+1 TO row
         ENDIF len(snolist&mi)>200
      ENDIF len(snolist&mi)>50
      RELEASE ALL LIKE snolist&mi
      STORE str(&mi+1,1,0) TO mi
   ENDDO WHILE &mi<=&no
   EJECT
   SET FORMAT TO SCREEN
   *    * increment bill number
   REPLACE billno WITH billno+1
   *    * store current date for next bill
   REPLACE FROM WITH mdate
   *    * store cumulative charges
   REPLACE ustdy WITH mec1,usequip WITH mec2,csgtdy WITH mec3
   REPLACE csgequip WITH mec4, csglbr WITH mec5

   STORE t TO c
   IF .not.b
      LOOP
   ENDIF .not.b
   SELECT SECOND
   USE pdf INDEX pdf.bn
   FIND '&mbn2'
   STORE bn TO mbn
   STORE sno TO msno
   *    * show user what battalion is being processed
   @ 12,0
   @ 12,25 SAY '   Battalion : ' GET mbn
   CLEAR GETS
ENDDO WHILE b
ERASE
RETURN
*
*       * END BNRPT.CMD
*
```

```
*******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: CLUSTERS.CMD
*
*       DESCRIPTION: This program allows the user to define "Project Reporting
*                    Groups," basically, sets of Schedule Numbers over which
*                    summary status reports are generated.  A "cluster" is a
*                    set of mutually exclusive Groups.  Maximum is nine
*                    clusters.
*
*       CALLS: GROUPS.CMD
*
*******************************************************************************
ERASE
SET COLON OFF
SET TALK OFF
SET DELETE ON
STORE "E" TO cmd

DO WHILE !(cmd)<> "Q"
   *    * set up input screen
   ERASE
@  1,21 SAY "CLUSTERS OF PROJECT REPORTING GROUPS"
@  3, 2 SAY "-----------------------------------------------------------------
-----------"
@  4, 2 SAY
"|
|"
@  5, 2 SAY "| 1>                        2>
3>                        |"
@  6, 2 SAY
"|
|"
@  7, 2 SAY
"|
|"
@  8, 2 SAY
"|
|"
@  9, 2 SAY "| 4>                        5>
6>                        |"
@ 10, 2 SAY
"|
|"
@ 11, 2 SAY
"|
|"
@ 12, 2 SAY
"|
|"
@ 13, 2 SAY "| 7>                     8>
9>                  |"
@ 14, 2 SAY
"|
|"
@ 15, 2 SAY
"|
|"
@ 16, 2 SAY
"|============================================================================|
"
@ 17, 2 SAY
"|
|"
@ 18, 2 SAY "|     E to edit/display/add a cluster      P to print
groups/schedule no.'s |"
@ 19, 2 SAY "|     D to delete a cluster                Q to quit to main
menu              |"
@ 20, 2 SAY
"|
|"
@ 21, 2 SAY "| OPTION ?"
@ 21,78 SAY "|"
@ 22, 2 SAY "-----------------------------------------------------------------
-----------"
   *    * display defined clusters
   USE clusters INDEX clusters
   GOTO TOP
   DO WHILE .NOT. EOF .AND. #<>0
      IF cno = 7
         STORE 2 TO row
      ELSE
         STORE int(cno/4) TO row
      ENDIF
      STORE cno-row*3 TO col
      @ ((row+1)*4)+1, 25*(col-1)+7 SAY cname
      SKIP
   ENDDO WHILE .NOT. EOF .AND. #<>0

   STORE 1 TO mcno
   STORE "E" TO cmd
   STORE "Y" TO mnew

   DO WHILE !(cmd)<> 'Q' .AND.!(cmd)<> 'B'
      *          * B used for control of call to GROUPS.CMD
      *          * get command from user
      @ 21, 13 GET cmd PICTURE "!"
      READ
```

```
CLEAR GETS
* edit or delete
IF !(cmd)= 'E' .OR.!(cmd)= 'D'
   @ 21,18 SAY "which cluster no.? (1-9)"
   @ 21,43 GET mcno PICTURE "9"
   READ
   CLEAR GETS
   STORE str(mcno,1,0) TO mkey
   IF mkey <> '0'
      FIND "&mkey"
      IF !(cmd)= 'E'
         IF #=0
            @ 21,48 SAY "adding new cluster? (y/n)"
            @ 21,74 GET mnew PICTURE "!"
            READ
            CLEAR GETS
            IF !(mnew)= 'Y'
               APPEND BLANK
               REPLACE cno WITH mcno
               IF cno = 7
                  STORE 2 TO row
               ELSE
                  STORE int(cno/4) TO row
               ENDIF cno = 7
               STORE cno-(row*3) TO col
                STORE "               Enter the cluster name.;"
               TO mnotel
               @ 21,15 GET mnotel
               CLEAR GETS
               @ ((row+1)*4+1),25*(col-1)+7 GET cname
               READ
               CLEAR GETS
               STORE cname TO mcname
               USE
               *********
               DO groups
               *********
               STORE "B" TO cmd
            ELSE
               IF !(mnew)= 'N'
                  @ 21,18 SAY "                                    "
               ENDIF !(mnew)= 'N'
            ENDIF !(mnew)= 'Y'
         ELSE
            STORE cname TO mcname
            USE
            **********
            DO groups
            **********
            STORE "B" TO cmd
         ENDIF #=0
      ELSE
```

```
               IF #=0
                  @ 21,48 SAY "No such cluster."
               ELSE
                  @ 21,48 SAY "sure? (y/n)     "
                  @ 21,61 GET mnew PICTURE "!"
                  READ
                  CLEAR GETS
                  IF !(mnew)= 'Y'
                     IF cno = 7
                        STORE 2 TO row
                     ELSE
                        STORE int(cno/4) TO row
                     ENDIF cno = 7
                     STORE cno-row*3 TO col
                     @ 21,18 SAY
"                                                            "
                     STORE " WAIT...PROCESSING  " TO mnote
                     @ ((row+1)*4+1),25*(col-1)+7 GET mnote
                     CLEAR GETS
                     DELETE
                     PACK
                     USE groups INDEX groups
                     DELETE FOR cno=mcno
                     PACK
                     USE schedule INDEX schedule
                     DELETE FOR cno=mcno
                     PACK
                     USE clusters INDEX clusters
                     @ ((row+1)*4+1),25*(col-1)+7 SAY "                     "
                     STORE "E" TO cmd
                     STORE 1 TO mcno
                  ELSE
                     @ 21,18 SAY
"                                                           "
                  ENDIF !(mnew)= 'Y'
               ENDIF #=0
            ENDIF !(cmd)= 'E'
         ENDIF mkey <> '0'
      ENDIF !(cmd)= 'E' .OR.!(cmd)= 'D'
      *
      *          * print out schedule numbers by groups
      IF !(cmd)= 'P'
         @ 21,18 SAY "which item no.? (1-9)"
         @ 21,41 GET mcno PICTURE "9"
         READ
         CLEAR GETS
         IF mcno>0.AND.mcno<9
            @ 21,18 SAY "                                          "
            STORE " WAIT...PREPARING REPORT FOR PRINT" TO mnote
            @ 21,20 GET mnote
            CLEAR GETS
            USE clusters INDEX clusters
            STORE str(mcno,1,0) TO mkey
            FIND "&mkey"
            IF #=0
               @ 21,18 SAY '
'
               STORE "Not a Project Reporting Cluster" TO mnote
               @ 21,48 GET mnote
               CLEAR GETS
            ELSE
               STORE cname TO mcname
               USE groups INDEX groups
               FIND '&mkey'
               IF EOF
                  @ 21,18 SAY '                                            '
                  STORE " No Groups in that Cluster" TO mnote
                  @ 21,48 GET mnote
                  CLEAR GETS
               ELSE
                  STORE gno TO mgno
                  STORE mkey+str(mgno,2,0) TO mkey2
                  STORE gname TO mgname
                  @ 21,18 SAY "                                           "
                  STORE " WAIT...PREPARING REPORT FOR PRINT" TO mnote
                  @ 21,20 GET mnote
                  CLEAR GETS
                  SELECT PRIMARY
                  USE schedule INDEX schedule
                  FIND '&mkey2'
                  STORE p.sno TO msno
                  SELECT secondary
                  USE pdf INDEX pdf
                  STORE " WAIT...PRINTING REPORT " TO mnote
                  @ 21,20 GET mnote
                  CLEAR GETS
                  SET FORMAT TO PRINT
                  DO WHILE (.NOT.EOF).AND.(p.cno=mcno).AND.(p.gno=mgno)
DO WHILE (.NOT.EOF).AND.(p.cno=mcno).AND.(p.gno=mgno)
   @ 2,30 SAY mcname
   @ 4,5 SAY "Schedule Numbers For Project Reporting Group:"
   @ 4,51 SAY mgname
   @ 6,5 SAY " Sch. No. Battalion          Title             Location"
   @ 7,5 SAY " ======== ========= ====================== ================="
   STORE 9 TO row
      DO WHILE (.NOT.EOF).AND.(row<62).AND.(p.gno=mgno).AND.(p.cno=mcno)
                           SELECT SECOND
                           FIND '&msno'
                           @ row,6 SAY s.sno
                           @ row,18 SAY s.bn
                           @ row,28 SAY s.title
                           @ row,54 SAY s.loc
                           STORE row+2 TO row
                           SELECT PRIMARY
                           SKIP +1
                           STORE p.sno TO msno
                        ENDDO
                        EJECT
                     ENDDO WHILE
                     SELECT PRIMARY
                     USE groups INDEX groups
                     FIND '&mkey2'
                     SKIP +1
                     IF EOF.OR.(p.cno<>mcno)
                        LOOP
                     ENDIF EOF.OR.(p.cno<>mcno)
                     STORE p.gno TO mgno
                     STORE p.gname TO mgname
                     STORE mkey+str(mgno,2,0) TO mkey2
                     SELECT PRIMARY
                     USE schedule INDEX schedule
                     FIND '&mkey2'
                     IF EOF.OR.(p.cno=mcno)
                        LOOP
                     ENDIF EOF.OR.(p.cno=mcno)
                     STORE p.sno TO msno
                     SELECT SECOND
                  ENDDO WHILE (.NOT.EOF).AND.(p.cno=mcno)
                  EJECT
                  SET FORMAT TO SCREEN
                  USE clusters INDEX clusters
                  @ 21,18 SAY
"                                                  "
                  STORE 'E' TO cmd
               ENDIF EOF
            ENDIF #=0
         ENDIF mcno>0.AND.mcno<9
        ENDIF cmd='P'
      ENDIF !(cmd)= 'E' .OR. !(cmd)= 'D'
      ENDDO

*       * quit with option to backup
IF !(cmd)= 'Q'
   STORE 'Y' TO mbak
   ERASE
   @ 13,0 SAY 'Do you want to backup Project Reporting Groups DataBase?' ;
   GET mbak PICTURE '!'
   @ 14,0 SAY '( You SHOULD if any proj. reporting group has been modified. )'
   READ
   IF !(mbak)= 'Y'
   ERASE
      ?
      ?
      ? 'WAIT.. COPYING PROJECT REPORTING GROUPS DATABASE TO B:'
      ?
      ?
      ?
      ?
      SELECT SECONDARY
      USE
      SELECT PRIMARY
      USE
      USE clusters
      COPY STRUCTURE TO b:clusters
      USE b:clusters
      INDEX ON STR(cno,1,0) TO b:clusters
      SET INDEX TO b:clusters
      APPEND FROM a:clusters
      USE groups
      COPY STRUCTURE TO b:groups
      USE b:groups
      INDEX ON STR(cno,1,0)+STR(gno,2,0) TO b:groups
      SET INDEX TO b:groups
      APPEND FROM a:groups
      USE schedule
      COPY STRUCTURE TO b:schedule
      USE b:schedule
      INDEX ON STR(cno,1,0)+STR(gno,2,0)+STR(xno,2,0) TO b:schedule
      SET INDEX TO b:schedule
      APPEND FROM a:schedule
      ERASE
      RETURN
   ENDIF !(mbak)= 'Y'
ENDIF !(cmd)= 'Q'
ENDDO WHILE
ERASE
RETURN
*
*       * END CLUSTERS.CMD
*
```

```
******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: COSTRPT.CMD
*
*       DESCRIPTION: Generates a report giving allocated versus expended
*                    funds by task.  The report is generated by
*                    clusters and will be subtotaled by group.
*
*       CALLS: None
*
******************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
SET DELETE ON
*       * get printer/console control codes
RESTORE FROM install ADDITIVE
*       * set up default memory variables
STORE 0 TO mcno
STORE 0 TO mgno
STORE "Y" TO ALL
*
USE misc
*       * get last processing date
STORE date TO mdate
*       * present clusters for user selection
@ 1,0 SAY "               DETAILED CONSTRUCTION COST REPORT"
@ 3,6 SAY "For period ending: " GET mdate PICTURE "999999"
CLEAR GETS
@ 5,0 SAY "Do you want to print out ALL the group reports for ALL of the "
@ 6,0 SAY "following clusters: (y/n) " GET ALL PICTURE '!'
*       * display clusters for user selection
USE clusters INDEX clusters
GOTO TOP
DO WHILE .NOT.EOF
   @ cno+7,2 SAY cno
   @ cno+7,3 SAY '>'
   @ cno+7,5 SAY cname
   SKIP
ENDDO WHILE .NOT.EOF
READ
CLEAR GETS

IF !(ALL)<> 'Y'
   @ cno+9,7 SAY "which cluster? " GET mcno PICTURE "9"
   @ cno+10,7 SAY "   0=EXIT"
   READ
   CLEAR GETS
   IF mcno=0
      RETURN
   ENDIF mcno=0
   STORE str(mcno,1,0) TO mkey
   FIND "&mkey"
   STORE cname TO mcna
   USE groups INDEX groups
   FIND "&mkey"
   IF #=0
      ? "                  NO Groups in Cluster"
      ? "                    Returning to Menu"
      STORE 1 TO count
      DO WHILE count<150
         STORE count+1 TO count
      ENDDO WHILE count<150
      RETURN
   ENDIF #=0
   USE schedule INDEX schedule
   COPY TO tsno FIELD cno,gno,sno FOR cno=mcno
ELSE
   USE schedule INDEX schedule
   COPY TO tsno FIELD cno,gno,sno
ENDIF !(ALL)<> 'Y'
@ 17,0
? ' Set Printer to Top Of Form'
? '  Press RETURN when ready'
WAIT
@ 16,0 SAY '                              '
@ 17,0 SAY '                              '
@ 18,0 SAY '                              '
@ 19,0 SAY '                              '
@ 23,30 SAY rv+'PROCESSING'+rvoff

*        * routine to convert mmddyy to military date: dd MMM yy
STORE 'JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC' TO year
STORE $(mdate,1,2) TO mnth
STORE $(mdate,3,2)+ ' ' +$(year,(val(mnth)*3-2),3)+ ' ' +$(mdate,5,2) TO mil
USE tsno
GOTO TOP
STORE sno TO msno
*        * routine to print report
SET FORMAT TO PRINT
SET MARGIN TO 0
*        * set print to 12 pitch
@ 0,0 SAY p12
DO WHILE .NOT.EOF
   STORE str(cno,1,0)+str(gno,2,0) TO mkey
   *    * variables mca? are for allocated funds
   STORE 0 TO mac1
   STORE 0 TO mac2
   STORE 0 TO mac3
   STORE 0 TO mac4
   STORE 0 TO mac5
   *    * variables mec? are for expended funds
   STORE 0 TO mec1
   STORE 0 TO mec2
   STORE 0 TO mec3
   STORE 0 TO mec4
   STORE 0 TO mec5
   SELECT SECOND
   USE groups INDEX groups
   FIND "&mkey"
   STORE gname TO mlgname
   @  1,28 SAY "DETAILED CONSTRUCTION COST REPORT"
   @  3, 1 SAY ep+mlgname+epoff
   @  3,80 SAY "AS OF"
   @  3,86 SAY ep+mil+epoff
   @ 4,33 SAY "Allocated/Expended Funds"
   @ 6,1 SAY "Sch. No. US TDY        US EQIUP       CSG TDY       CGS
EQUIP     CSC LABOR     TOTAL"
   @ 7,1 SAY "======== ============  ============  ============
============  ============  ============"
   SELECT SECONDARY
   USE ps INDEX ps
   STORE 9 TO row
   SELECT PRIMARY
   STORE cno TO mcno
   STORE gno TO mgno

   DO WHILE p.gno=mgno .AND. p.cno=mcno .AND. (.NOT.EOF)
      IF row>=63
         EJECT
         @  1,29 SAY "DETAILED CONSTRUCTION COST REPORT"
         @  3, 0 SAY ep+mlgname+epoff
         @  3,75 SAY "AS OF"
         @  3,81 SAY ep+mil+epoff
         @ 4,33 SAY "Allocated/Expended Funds"
         @ 6,1 SAY "Sch. No. US TDY        US EQIUP       CSG TDY       CGS
EQUIP     CSC LABOR     TOTAL"
         @ 7,1 SAY "======== ============  ============  ============
============  ============  ============"
         STORE 9 TO row
      ENDIF row>=63
            SELECT SECONDARY
            FIND '&msno'
            @ row,1 SAY s.sno
            @ row,10 SAY s.ac1 USING '9,999,999.99'
            @ row,24 SAY s.ac2 USING '9,999,999.99'
            @ row,38 SAY s.ac3  USING '9,999,999.99'
            @ row,52 SAY s.ac4  USING '9,999,999.99'
            @ row,66 SAY s.ac5  USING '9,999,999.99'
            @ row,80 SAY s.ac1+s.ac2+s.ac3+s.ac4+s.ac5  USING '99,999,999.99'
            STORE s.ac1+mac1 TO mac1
            STORE s.ac2+mac2 TO mac2
            STORE s.ac3+mac3 TO mac3
            STORE s.ac4+mac4 TO mac4
            STORE s.ac5+mac5 TO mac5
            STORE row+1 TO row
            @ row,10 SAY s.ec1  USING '9,999,999.99'
            @ row,24 SAY s.ec2  USING '9,999,999.99'
            @ row,38 SAY s.ec3  USING '9,999,999.99'
            @ row,52 SAY s.ec4  USING '9,999,999.99'
            @ row,66 SAY s.ec5  USING '9,999,999.99'
            @ row,80 SAY s.ec1+s.ec2+s.ec3+s.ec4+s.ec5  USING '99,999,999.99'
            STORE s.ec1+mec1 TO mec1
            STORE s.ec2+mec2 TO mec2
            STORE s.ec3+mec3 TO mec3
            STORE s.ec4+mec4 TO mec4
            STORE s.ec5+mec5 TO mec5
            STORE row+2 TO row
            SELECT PRIMARY
            SKIP +1
            STORE p.sno TO msno

   ENDDO WHILE gno=mgno .AND. cno=mcno .AND. (.NOT.EOF)
   @ row,1 SAY "======== ============  ============  ============
============  ============  ============"
   STORE row+1 TO row
   @ row,1 SAY 'TOTAL'
   @ row,10 SAY mac1  USING '9,999,999.99'
   @ row,24 SAY mac2  USING '9,999,999.99'
   @ row,38 SAY mac3  USING '9,999,999.99'
   @ row,52 SAY mac4  USING '9,999,999.99'
   @ row,66 SAY mac5  USING '9,999,999.99'
   @ row,80 SAY mac1+mac2+mac3+mac4+mac5  USING '99,999,999.99'
   STORE row+1 TO row
   @ row,10 SAY mec1  USING '9,999,999.99'
   @ row,24 SAY mec2  USING '9,999,999.99'
   @ row,38 SAY mec3  USING '9,999,999.99'
   @ row,52 SAY mec4  USING '9,999,999.99'
   @ row,66 SAY mec5  USING '9,999,999.99'
   @ row,80 SAY mec1+mec2+mec3+mec4+mec5  USING '99,999,999.99'

   EJECT
ENDDO WHILE .NOT.EOF
*       * set printer back to 10 pitch
@ 0,0 SAY p10
SET FORMAT TO SCREEN
USE
DELETE FILE tsno
ERASE
RETURN
*
*       * END COSTRPT.CMD
```

```
******************************************************************************
*
*                         A - C - M - S  Ver. 84.0
*
*                                 USA-CERL
*
*       PROGRAM: EDEQ.CMD
*
*       DESCRIPTION: This is the data entry routine for the equipment cost
*                    data base EQCOST.CMD.  Calls EDEQA.CMD to enter
*                    attachments.  Inherits memory variable meno (item number)
*                    and cursor.
*
*
*       CALLS: called from EQCOST.CMD and calls EDEQA.CMD
*
******************************************************************************
ERASE
STORE " " TO mtype
*       * types of equipment are: U for US,L for local purchase
*       * S for special purchase
DO WHILE !(mtype)<> "U" .AND.!(mtype)<> "L" .AND.!(mtype)<> "S"
   @ 1,19 SAY "HOURLY RATES OF EQUIPMENT COSTS"
   @ 2, 0 SAY "ITEM NO:"
   @ 2, 9 SAY eno
   @ 2,13 SAY "."
   @ 2,14 SAY atchno
   IF !(type)<> "U" .AND.!(type)<> "L" .AND.!(type)<> "S"
      @ 3,33 SAY "Illegal type->"
   ENDIF !(type)<> "U" .AND.!(type)<> "L" .AND.!(
   @ 3,49 SAY "TYPE"
   @ 3,54 GET type
   @ 3,59 SAY "(U,L, or S)"
   @ 4, 0 SAY "NOMEMCLATURE :"
   @ 4,14 GET desp1
   @ 4,44 SAY ":       U - US STANDARD EQUIP"
   @ 5,13 SAY ":"
   @ 5,14 GET desp2
   @ 5,44 SAY ":       L - LOCAL & COMMERCIAL"
   @ 6,13 SAY ":"
   @ 6,14 GET desp3
   @ 6,29 SAY ":                      S - EDU SPECIAL PURCHASED"
   @ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field;
     ;  CTR-D: next character ;"
   @ 23, 5 SAY "CTR-W: next screen ;  CTR-S: previous character ;"
   READ
   CLEAR GETS
   REPLACE type WITH !(type)
   STORE type TO mtype
ENDDO WHILE !(mtype)<> "U" .AND.!(mtype)<> "L" .AND.!(mtype)<> "S"
```

EDEQ.CMD

```
STORE 1 TO x
@ 8, 0 SAY "FSN :"
@ 8, 5 GET a
@ 8, 9 SAY "-"
@ 8,10 GET b
@ 8,12 SAY "-"
@ 8,13 GET c
@ 8,16 SAY "-"
@ 8,17 GET d
@ 8,21 SAY ":"
*       *  skips leading blanks to left-adjust the equip
*       * description so that they
*       * can be alphabetized correctly.
DO WHILE $(despl,x,1)= " " .AND.x<=29
   STORE x+1 TO x
ENDDO WHILE $(despl,x,1)= " " .AND.x<=29
REPLACE despl WITH $(despl,x,30)
@ 10, 0 SAY "ACQUISITION COST :"
@ 10,18 GET acost
@ 10,25 SAY ".00:     Not on AMDF?"
@ 10,47 GET f
@ 10,49 SAY "(type True or False)"
READ
CLEAR GETS
IF !(f)= 'T'
   @ 10,45
ELSE
   @ 10,33
ENDIF !(f)= 'T'
CLEAR GETS
@ 12, 0 SAY "DEPRECIATION                         OPERATION & MAINTEN."
@ 13, 0 SAY "COST (unfunded)   REPORTABLE? (y/n)"
@ 13,35 GET r
@ 13,38 SAY "COST  (funded)"
READ
CLEAR GETS
IF !(r)= 'N'
   @ 13,18 SAY "NOT reportable    "
   @ 13,54 SAY "NOT reportable"
ELSE
   @ 13,18 SAY ":  .00:            "
   @ 13,54 SAY ":  .00:"
   @ 13,19 GET dcost
   @ 13,55 GET omcost
   READ
ENDIF !(r)= 'N'
CLEAR GETS
STORE 'N' TO mok
STORE 'A' TO m
*       * case=2 from EQCOST.CMD is change equipment
*       * so this option not needed
IF mcmd<>'2'
```

```
   @ 21,8 SAY "Want to add Attachments, or Individual components?" ;
   GET mok PICTURE '!'
   READ
   @ 21,0
IF mok<>'Y'
   REPLACE n WITH 'N'
ELSE
   DO WHILE (!(m)<> 'A') .AND.(!(m)<> 'I')
      @ 19,60 SAY "(type A, or I, ): " GET m PICTURE '!'
      READ
      STORE 'Y' TO mok
   ENDDO WHILE
   @ 19,0
   REPLACE n WITH m
   STORE 0 TO matchno
   STORE str(eno,4,0) TO ceno
   FIND "&ceno"
   *    * determine the next attachment number available
   DO WHILE .NOT.EOF.AND.eno=meno
      IF atchno>matchno
         STORE atchno TO matchno
      ENDIF atchno>matchno
      SKIP 1
   ENDDO WHILE .NOT.EOF.AND.eno=meno
   STORE matchno+1 TO matchno
   DO WHILE !(mmore)= 'Y'
   STORE ceno+str(matchno,2,0) TO mact
      APPEND BLANK
      REPLACE eno WITH meno, atchno WITH matchno, type WITH mtype
      REPLACE r WITH 'Y' , n WITH m, f WITH 'F'
      CLEAR GETS
      *         * add attachments
      ********
      DO edeqa
      ********
      *         * increment attachment number
      STORE matchno+1 TO matchno
    ENDDO WHILE !(mmore)= 'Y'
   ENDIF
   *     * if user is specifing equipment number (eno)
   *     * then go back to main EQCOST.CMD
   IF !(spec)<>'Y'
     @ 21,8 SAY " Continue adding NEW equipment? (y/n)" GET mmore PICTURE '!'
     READ
     CLEAR GETS
     @ 21,0
     CLEAR GETS
     RETURN
   ENDIF
 ENDIF
 RETURN
 *
```

```
*       * END EDEQ.CMD
*
```

```
******************************************************************************
*
*                          A - C - M - S  Ver. 84.0
*
*                                 USA-CERL
*
*       PROGRAM: EDEQA.CMD
*
*       DESCRIPTION: Lets user add equipment information about
*                    attachments or individual components.  Displays
*                    parent equipment data on upper part of screen
*                    for user's benefit.
*
*       CALLS: none; called by EQCOST.CMD and EDEQ.CMD
*
******************************************************************************
ERASE
IF !(n)<>'I'
   *    * default is 'attachments'
   STORE "----    ATTACHMENTS    --------" TO mname
ELSE
   STORE "---  INDIVIDUAL COMPONENTS  -" TO mname
ENDIF
* find parent equipment for this attachment
FIND '&ceno'
@  2, 0 SAY "ITEM NO:"
@  2, 9 SAY eno
@  2,13 SAY "."
@  2,14 SAY atchno
@  4, 0 SAY "NOMENCLATURE :"
@  4,14 SAY desp1
@  4,44 SAY ":     TYPE:"
@  4,56 SAY type
@  5,13 SAY ":"
@  5,14 SAY desp2
@  5,44 SAY ":"
@  6,13 SAY ":"
@  6,14 SAY desp3
@  6,29 SAY ":"
@  8, 0 SAY "------------------&mname-----------------------------"
* find the blank record already created for this attachment
FIND '&mact'
@ 10, 0 SAY "ITEM NO:"
@ 10, 9 SAY eno
@ 10,13 SAY "."
@ 10,14 SAY atchno
@ 12, 0 SAY "NOMEMCLATURE :"
@ 12,14 GET desp1
@ 12,44 SAY ":     TYPE:"
@ 12,56 SAY type
@ 13,13 SAY ":"
@ 13,14 GET desp2
@ 13,44 SAY ":"
@ 14,13 SAY ":"
@ 14,14 GET desp3
@ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field;
     ;  CTR-D: next character ;"
@ 23, 5 SAY "CTR-W: next screen ;  CTR-S: previous character ;"
READ
@ 14,29 SAY ":"
CLEAR GETS
@ 14,50 SAY "FSN :"
@ 14,55 GET A
@ 14,59 SAY "-"
@ 14,60 GET B
@ 14,62 SAY "-"
@ 14,63 GET C
@ 14,67 SAY "-"
@ 14,68 GET D
@ 14,72 SAY ":"
*       * skip any leading blanks of the equipment description
*       * so we can alphabetize
STORE 1 TO x
DO WHILE $(desp1,x,1)=' '.AND.x<=29
   STORE x+1 TO x
ENDDO
REPLACE desp1 WITH $(desp1,x,30)
@ 16, 0 SAY "ACQUISITION COST :"
@ 16,18 GET ACOST
@ 16,25 SAY ".00:     Not on AMDF?"
@ 16,47 GET f
@ 16,49 SAY "(type True or False)"
READ
IF !(f)='T'
   @ 16,45
ELSE
   @ 16,33
ENDIF
CLEAR GETS
@ 18, 0 SAY "DEPRECIATION                              OPERATION & MAINTEN."
@ 19, 0 SAY "COST (unfunded)   REPORTABLE? (y/n)"
@ 19,35 GET r
@ 19,38 SAY "COST  (funded)"
READ
CLEAR GETS
IF !(r)='N'
   @ 19,18 say"NOT reportable     "
   @ 19,54 say "NOT reportable"
ELSE
   @ 19,18 say ":  .00:           "
   @ 19,54 say ":  .00:"
   @ 19,19 get DCOST
   @ 19,55 get OMCOST
```

```
READ
ENDIF
CLEAR GETS
*       * case=2 from EQCOST.CMD for change equipment
*       * so skip this part of program
IF mcmd<>'2'
   @ 21,0
   @ 21, 0 SAY "Continue adding attachments or individule components? (y/n)";
   GET mmore PICTURE '!'
   READ
   CLEAR GETS
ENDIF
RETURN
*
*       * END EDEQA.CMD
*
```

```
*****************************************************************************
*
*                          A - C - M - S  Ver. 84.0
*
*                               USA-CERL
*
*       PROGRAM: EQAUDIT.CMD
*
*       DESCRIPTION: Generates weekly equipment cost for task by item.
*                    Lists description, item number, $/hr, hours used, cost
*                    and total cost.  The data base EQ.DBF stores old weekly
*                    equipment data.  EQ.DBF is updated by BACKUP.CMD.
*
*
*       CALLS: None
*
*****************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
SET DELETE ON
SELECT SECONDARY
USE eq INDEX eq
*       * get printer/console control codes
RESTORE FROM install ADDITIVE
*       * abbreviations for months
STORE 'JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC' TO year
STORE 'A' TO stop
DO WHILE stop ='A'
STORE '      ' TO msno
@ 1,10 SAY "      GRAF84 EQUIPMENT COST SUMMARY REPORT"
@ 6,10 SAY "           Enter Schedule Number: " get msno PICTURE '999999'
@ 7,10 SAY "             or RETURN to EXIT"
READ
CLEAR GETS
IF msno='      '
   STORE 'B' TO stop
   LOOP
ENDIF
FIND '&msno'
IF #=0
@ 7,0
??  "                NO EQUIPMENT REPORTED"
LOOP
ENDIF
*       * get earliest date
STORE week TO mweek1
DO WHILE (sno=msno).and.(.not.eof)
  SKIP +1
ENDDO
SKIP-1
```

```
*       * get last date
STORE week TO mweek2
@ 9,10 SAY '           FROM: ' GET mweek1 PICTURE '999999'
@ 10,10 SAY '              TO: ' GET mweek2 PICTURE '999999'
READ
CLEAR GETS
@ 15,30 SAY rv+ 'PROCESSING' +rvoff
STORE mweek1 TO mdate
STORE $(mdate,1,2) TO mnth
STORE $(mdate,3,2)+ ' ' +$(year,(val(mnth)*3-2),3)+ ' ' +$(mdate,5,2) TO mill
STORE mweek2 TO mdate
STORE $(mdate,1,2) TO mnth
STORE $(mdate,3,2)+ ' ' +$(year,(val(mnth)*3-2),3)+ ' ' +$(mdate,5,2) TO mil

SELECT PRIMARY
*       * get report header
USE pdf INDEX pdf
FIND "&msno"
*       * go through all records for periods specified
*       * printing equipment use data
SET FORMAT TO PRINT
STORE 12 TO row
@ 1,4 SAY "GRAF84 Summary Equipment Cost Report From:&mill  To:&mil"
@ 3, 6 SAY "Schedule No. :"
@ 3,20 SAY sno
@ 3,43 SAY "Title     :"
@ 3,54 SAY title
@ 4, 6 SAY "Battalion    :"
@ 4,20 SAY bn
@ 4,43 SAY "Location  :"
@ 4,54 SAY loc
@ 6,0 SAY
'*****************************************************************************
'
@ 8,0 SAY '      DESCRIPTION                ITEM NO.    $/HOUR
HOURS         COST  '
@ 10,0 SAY '-----------------------------------------------------------------
-----------'
SELECT PRIMARY
USE eqcost INDEX eqcost
GOTO TOP
SELECT SECOND
FIND "&msno&mweek1"
IF #=0
   @ row,20 SAY "NO EQUIPMENT REPORTED FOR THIS TASK"
   EJECT
   GO BOTTOM
   SKIP +1
   LOOP
ENDIF #=0
STORE 0 TO mtotal
DO WHILE (sno=msno).AND.(.NOT.EOF).AND.(week<=mweek2)
```

```
   STORE 0.0 TO mitem
   STORE 0.0 TO msub
   STORE week TO mdate
   STORE msno+week TO msnoweek
   STORE $(mdate,1,2) TO mnth
   STORE $(mdate,3,2)+ ' ' +$(year,(val(mnth)*3-2),3)+ ' ' +$(mdate,5,2) TO
mil
   DO WHILE (sno+week=msnoweek).AND.(.NOT.EOF).AND.#<>0
      STORE "1" TO no
      DO WHILE no<= "4"
         IF (!(s.type&no)= "O").OR.(p.eno=0).OR.(s.eno&no=0)
            STORE str(&no+1,1,0) TO no
            LOOP
         ENDIF s.type&no= "O" .OR.(p.eno=0)
         STORE str(s.eno&no,4,0)+str(s.atchno&no,2,0) TO mkey
         SELECT PRIMARY
         FIND '&mkey'
         IF #=0
            STORE str(&no+1,1,0) TO no
            LOOP
         ENDIF s.eno&no=0
         STORE p.omcost*s.hr&no TO mitem
         @ row,2 SAY p.despl
         IF s.atchno&no=0
            @ row,35 SAY str(s.eno&no,4,0)+'.0'
         ELSE
             @ row,35 SAY s.eno&no+(s.atchno&no/10.0) USING '9999.99'
         ENDIF
         @ row,45 SAY p.omcost USING '$999.99'
         @ row,57 SAY s.hr&no
         @ row,65 SAY mitem USING '$9999999.99'
         STORE row+2 TO row
         STORE mitem+msub TO msub
         STORE str(&no+1,1,0) TO no
      ENDDO WHILE no<= "4"
      SELECT SECOND
      SKIP
   ENDDO WHILE (sno+week=msnoweek).AND.(.NOT.EOF)
   @ row,0 SAY '--------------------------------------------------------------
---------------'
   STORE row+2 TO row
   @ row,31 SAY 'WEEKLY SUBTOTAL FOR '+'&mil'
   @ row,63 say msub USING '$999999999.99'
   STORE row+2 TO row
   @ row,0 SAY '--------------------------------------------------------------
---------------'
   STORE row+2 TO row
   STORE mtotal+msub TO mtotal
 ENDDO
STORE row+1 to row
@ row,0 SAY
'=============================================================================
'
@ row+2,15 SAY '      TOTAL '+'&mill'+' TROUGH '+'&mil'
@ row+2,54 SAY mtotal USING  '$999999999.99'
EJECT
SET FORMAT TO SCREEN
ERASE
ENDDO
ERASE
RETURN
*
*       * END EQAUDIT.CMD
*
```

```
*******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: EQCOST.CMD
*
*       DESCRIPTION: Presents main menu for equipment cost data base.
*                    Gives user options to enter new equipment, change,
*                    delete, or print listing of all equipment.  The program
*                    will assign a unique number 'eno' to each new equipment
*                    item, but user may specify.  Items may have attachments.
*                    Primary use of this data is for calculating task
*                    equipment use charges.
*
*
*       CALLS: EDEQ.CMD, EDEQA.CMD, and EQRPT.CMD
*
*******************************************************************************
SET COLON OFF
SET TALK OFF
SET DELETE ON
STORE '1' TO mcmd

DO WHILE mcmd<> '0'
   ERASE
   STORE 'Y' TO mmore
   @  1, 8 SAY
"**************************************************************"
   @  2, 8 SAY "*
*"
   @  3, 8 SAY "*        EQUIPMENT COST DATABASE MENU
*"
   @  4, 8 SAY "*
*"
   @  5, 8 SAY
"**************************************************************"
   @  6, 8 SAY "*
*"
   @  7, 8 SAY "*
*"
   @  8, 8 SAY "*    <1> ADD NEW EQUIPMENT
*"
   @  9, 8 SAY "*
*"
   @ 10, 8 SAY "*    <2> CHANGE EQUIPMENT COST DATA
*"
   @ 11, 8 SAY "*
*"
```

```
   @ 12, 8 SAY "*    <3> DELETE EQUIPMENT OR ATTACHMENTS
*"
   @ 13, 8 SAY "*
*"
   @ 14, 8 SAY "*    <4> PRINT EQUIPMENT COST CATALOG
*"
   @ 15, 8 SAY "*
*"
   @ 16, 8 SAY "*    <0> EXIT
*"
   @ 17, 8 SAY "*
*"
   @ 18, 8 SAY "*
*"
   @ 19, 8 SAY
"**************************************************************"
   @ 21,8 SAY "OPTION? " GET mcmd
   READ
   CLEAR GETS

   SELECT PRIMARY
   *    * use eqcost
   *    * index is str(eno,4,0)+str(atchno,2,0) to eqcost.ndx
   *    * eqcost.dbf stores all the equipment and attchments
   USE eqcost INDEX eqcost

   DO CASE

      CASE mcmd= '1'
         *      * add new equipment
         STORE 'N' TO spec
         @ 21,0
         @ 21,8 SAY 'Do you want to specify ITEM NO. ? (y/n)' GET spec;
         PICTURE '!'
         READ
         IF !(spec)='Y'
            STORE 0 TO meno
            STORE 0 TO matchno
            @ 21,0
            @ 21,8  SAY "Enter the ITEM NO. :    .  :"
            @ 21,28 GET meno PICTURE "9999"
            @ 21,33 GET matchno PICTURE "99"
            READ
            CLEAR GETS
            STORE str(meno,4,0)+str(matchno,2,0) TO mkey
            FIND '&mkey'
            IF #<>0
              @ 21,0
              @ 21,8 SAY "Item already in data base-- Press RETURN"
              SET CONSOLE OFF
              WAIT
              SET CONSOLE ON
```

```
        @ 21,0
        LOOP
      ELSE
        IF matchno=0
           APPEND BLANK
           REPLACE eno WITH meno,atchno WITH 0
           REPLACE r WITH 'y' , f WITH 'f' ,type WITH 'U' ,n WITH 'N'
           *******
           DO edeq
           *******
        ELSE
           STORE STR(meno,4,0)+' 0' TO mkey2
           FIND '&mkey2'
           IF #=0
              @ 21,0
              @ 21,8 SAY 'NO ITEM to add attachments to!--Press RETURN'
              SET CONSOLE OFF
              WAIT
              SET CONSOLE ON
              LOOP
           ENDIF
           STORE type TO mtype
           APPEND BLANK
           REPLACE eno WITH meno,atchno WITH matchno,type WITH mtype
           REPLACE r WITH 'Y',n WITH 'A',f WITH 'F'
           STORE STR(meno,4,0) TO ceno
           STORE mkey TO mact
           FIND '&mact'
           *       * add attachments
           ********
           DO edeqa
           ********
        ENDIF
      ENDIF
   ENDIF

 DO WHILE (!(mmore)='Y').AND.(!(spec)<>'Y')
   GO BOTTOM
   *      * new item number is meno
   STORE eno+1 TO meno
   APPEND BLANK
   REPLACE eno WITH meno.atchno WITH 0
   REPLACE r WITH 'y' , f WITH 'f' ,type WITH 'U' ,n WITH 'N'
   *******
   DO edeq
   *******
 ENDDO WHILE !(mmore)= 'Y'


CASE mcmd= '2'
   *      * change equipment
   STORE 0 TO meno
   STORE 0 TO matchno
   @ 21,0
   @ 21,8  SAY "Enter the ITEM NO. :    .  :"
   @ 21,28 GET meno PICTURE "9999"
   @ 21,33 GET matchno PICTURE "99"
   READ
   CLEAR GETS
   STORE str(meno,4,0)+str(matchno,2,0) TO mkey
   FIND '&mkey'
   IF #=0
     @ 21,0
     @ 21,8 SAY "Item NOT Found -- Press RETURN"
     SET CONSOLE OFF
     WAIT
     SET CONSOLE ON
     @ 21,0
   ELSE
     IF matchno=0
        *******
        DO edeq
        *******
     ELSE
        STORE STR(meno,4,0) TO ceno
        STORE mkey TO mact
        FIND '&mact'
        ********
        DO edeqa
        ********
     ENDIF
   ENDIF

CASE mcmd= '3'
   *      * delete equipment
   STORE 0 TO meno
   STORE 0 TO matchno
   @ 21,0 SAY "Enter the item no. :    .  :    "
   @ 21,20 GET meno PICTURE "9999"
   @ 21,25 GET matchno PICTURE "99"
   READ
   CLEAR GETS
   STORE str(meno,4,0)+str(matchno,2,0) TO mkey
   FIND '&mkey'
   IF #=0
     @ 21,0
     @ 21,8 SAY "Item NOT Found -- Press RETURN"
     SET CONSOLE OFF
     WAIT
     SET CONSOLE ON
     @ 21,0
   ELSE
     IF matchno=0
       DELETE ALL FOR eno=meno
```

```
           ELSE
             DELETE
           ENDIF
         ENDIF

      CASE mcmd= '4'
         *      * print equipment listing
         ********
         DO eqrpt
         ********

      CASE mcmd= '0'
         *      * exit program with option to backup to B drive
         STORE 'Y' TO ok
         @ 21,8 SAY "Do you want to backup the equipment database to B: ?
(y/n)";
         GET ok PICTURE '!'
         READ
         IF !(ok)= "Y"
         ERASE
         ?
         ?
         ?
         ? 'WAIT..COPYING EQUIPMENT COST DATA TO B:'
         ?
         ?
         ?
         ?
         SELECT PRIMARY
         USE
         USE eqcost
         COPY STRUCTURE TO b:eqcost
         USE b:eqcost
         INDEX ON STR(eno,4,0)+STR(atchno,2,0) TO b:eqcost
         SET INDEX TO b:eqcost
         APPEND FROM a:eqcost
         ERASE
         QUIT
         ENDIF !(ok)= "Y"
         ERASE
         RETURN

      OTHERWISE

   ENDCASE

ENDDO WHILE mcmd<> '0'
*
*       * END EQCOST.CMD
*
```

```
******************************************************************************
*
*                         A - C - M - S  Ver. 84.0
*
*                                USA-CERL
*
*       PROGRAM: EQRPT.CMD
*
*       DESCRIPTION: Routine to generate the Equipment Cost Catalog.  Report
*                    is alphabetized on description and divided into three
*                    sections: US, Local Purchase, and EUD Special Equipment.
*                    Uses temporary index files EQCOST.ABC and AEQTEMP.ABC
*                    as alphabetical index.  Also creates AEQTEMP.DBF as
*                    temporary data base for attachments.  Temporary files
*                    are deleted on termination of report.
*
*       CALLS: Called From EQCOST.CMD
*
******************************************************************************
@ 21,0
STORE 'WAIT...PREPARING REPORT' TO mnote
@ 21,8 GET mnote
CLEAR GETS
*       * clear primary use data base: eqcost
SELECT PRIMARY
USE
*       * select as second
SELECT SECOND
USE eqcost
*       * reindex it
INDEX ON str(eno,4,0)+str(atchno,2,0) TO eqcost
SET INDEX TO eqcost
*       * create temporary data base of parent equipment
COPY TO teq FOR atchno=0
SELECT PRIMARY
USE teq
*       * index on equipment type and in alphabetical order
INDEX ON !(type)+!(despl) TO teq
SET INDEX TO teq
@ 21,0
STORE 'WAIT...PRINTING REPORT' TO mnote
@ 21,8 GET mnote
CLEAR GETS
SET FORMAT TO PRINT
SET MARGIN TO 0
*       * get printer/console control codes
RESTORE FROM install ADDITIVE
STORE 1 TO page
STORE 0 TO row
*       * print report header
@ 0,3 SAY ep+dw+ "HOURLY RATES OF EQUIPMENT COSTS" +p12
@ 4,55 SAY "DEPRECIATION"
@ 4,70 SAY "OPERATION &"
@ 5,0 SAY "ITEM"
@ 5,13 SAY "NOMENCLATURE"
@ 5,40 SAY "ACQUISITION"
@ 5,55 SAY "COST"
@ 5,70 SAY "MAINTENANCE"
@ 6,0 SAY "NO."
@ 6,13 SAY "   & FSN"
@ 6,40 SAY "COST"
@ 6,55 SAY "(Unfunded)"
@ 6,70 SAY "COST (Funded)&epoff"

STORE 9 TO row
*       * find first special equipment
FIND S
*       * store its record number
STORE # TO eudmark
*       * find first US equipmnet
FIND U
*       * store record number
STORE # TO stop
STORE str(eno,4,0) TO meno
STORE eno TO meno2
STORE ' ' TO mnote
STORE 'T' TO mmore
DO WHILE mmore='T'
   *    * print routine for each item
   IF row>54
      @ 58,41 SAY page
      STORE page+1 TO page
      EJECT
      STORE 6 TO row
        @ 0,20 SAY ep+mnote
        @ 1,55 SAY "DEPRECIATION"
        @ 1,70 SAY "OPERATION &"
        @ 2,0 SAY "ITEM"
        @ 2,13 SAY "NOMENCLATURE"
        @ 2,40 SAY "ACQUISITION"
        @ 2,55 SAY "COST"
        @ 2,70 SAY "MAINTENANCE"
        @ 3,0 SAY "NO."
        @ 3,13 SAY "   & FSN"
        @ 3,40 SAY "COST"
        @ 3,55 SAY "(Unfunded)"
        @ 3,70 SAY "COST (Funded)&epoff"

   ENDIF row>54

   IF atchno=0
      @ row-1,0 SAY ep
      @ row,0 SAY str(eno,3,0)+ ".0"
```

```
ELSE
   @ row,0 SAY str(eno,3,0)+ "."
   IF atchno>=10
      @ row,4 SAY atchno USING "99"
   ELSE
      @ row,4 SAY atchno USING "9"
   ENDIF atchno>=10
ENDIF atchno=0

@ row,7 SAY despl
@ row,40 SAY acost USING "$999,999.99"

IF !(r)= 'Y'
   @ row,58 SAY dcost USING "$99.99"
   @ row,72 SAY omcost USING "$99.99"
ELSE
   @ row,58 SAY "(  NOT REPORTABLE  )"
ENDIF
   @ row,79 SAY epoff
   @ row+1,7 SAY desp2
   IF !(f)= 'T'
      @ row+1,40 SAY "(Not on AMDF)"
   ENDIF !(f)= 'T'
   STORE row+2 TO row
   IF desp3<> "        "
      @ row,7 SAY desp3
      STORE row+1 TO row
   ENDIF desp3<> "      "

   IF a<> "    " .AND.b<> "  " .AND.c<> "   " .AND.d<> "    "
      @ row,7 SAY a+ '-' +b+ '-' +c+ '-' +d
      STORE row+1 TO row
   ENDIF a<> "    " .AND.b<> "  " .AND.c<> "   "
   STORE row+1 TO row
   STORE n TO m

   *         * look for attachments
   SELECT SECOND
   FIND "&meno"
   SKIP
   IF atchno<>0
     STORE 1 TO title
     DO WHILE (meno2=eno).AND.(.NOT.EOF)
         IF (!(m)='A').AND.(title=1)
            @ row,5 SAY "ATTACHMENTS:"
            STORE 0 TO title
         ELSE
            IF (!(m)='I').AND.(title=1)
               @ row,5 SAY "INDIVIDULE COMPONENTS:"
               STORE 0 TO title
            ENDIF !(m)='I'
         ENDIF !(m)='A'
```

```
STORE row+2 TO row
IF row>54
   @ 58,41 SAY page
   STORE page+1 TO page
   EJECT
   STORE 6 TO row
   @ 0,20 SAY ep+mnote
   @ 1,55 SAY "DEPRECIATION"
   @ 1,70 SAY "OPERATION &"
   @ 2,0 SAY "ITEM"
   @ 2,13 SAY "NOMENCLATURE"
   @ 2,40 SAY "ACQUISITION"
   @ 2,55 SAY "COST"
   @ 2,70 SAY "MAINTENANCE"
   @ 3,0 SAY "NO."
   @ 3,13 SAY "   & FSN"
   @ 3,40 SAY "COST"
   @ 3,55 SAY "(Unfunded)"
   @ 3,70 SAY "COST (Funded)&epoff"
ENDIF row>50

IF atchno=0
   @ row-1,0 SAY ep
   @ row,0 SAY str(eno,3,0)+ ".0"
ELSE
   @ row,0 SAY str(eno,3,0)+ "."
   IF atchno>=10
       @ row,4 SAY atchno USING "99"
   ELSE
       @ row,4 SAY atchno USING "9"
   ENDIF atchno>=10
ENDIF atchno=0

@ row,7 SAY desp1
@ row,40 SAY acost USING "$999,999.99"

IF !(r)= 'Y'
   @ row,58 SAY dcost USING "$99.99"
   @ row,72 SAY omcost USING "$99.99"
ELSE
   @ row,58 SAY "(  NOT REPORTABLE  )"
ENDIF
   @ row,79 SAY epoff
   @ row+1,7 SAY desp2
IF !(f)= 'T'
   @ row+1,40 SAY "(Not on AMDF)"
ENDIF !(f)= 'T'

STORE row+2 TO row
IF desp3<> "      "
   @ row,7 SAY desp3
   STORE row+1 TO row
```

```
      ENDIF desp3<> "      "

      IF a<> "    " .AND.b<> "  " .AND.c<> "   " .AND.d<> "    "
         @ row,7 SAY a+ '-' +b+ '-' +c+ '-' +d
         STORE row+1 TO row
      ENDIF a<> "   " .AND.b<> "  " .AND.c<> "   "
      STORE row+1 TO row
      SKIP

  ENDDO WHILE

ENDIF atchno<>0

SELECT PRIMARY

SKIP
IF EOF
   @ 58,40 SAY page
   STORE page+1 TO page
   EJECT
   STORE ep+ "LOCAL PURCHASED AND US COMMERCIAL EQUIPMENT" TO mnote
   STORE 6 TO row
   @ 0,20 SAY mnote
   @ 1,55 SAY "DEPRECIATION"
   @ 1,70 SAY "OPERATION &"
   @ 2,0 SAY "ITEM"
   @ 2,13 SAY "NOMENCLATURE"
   @ 2,40 SAY "ACQUISITION"
   @ 2,55 SAY "COST"
   @ 2,70 SAY "MAINTENANCE"
   @ 3,0 SAY "NO."
   @ 3,13 SAY "   & FSN"
   @ 3,40 SAY "COST"
   @ 3,55 SAY "(Unfunded)"
   @ 3,70 SAY "COST (Funded)&epoff"

   GOTO TOP
ENDIF EOF

IF #=eudmark
   @ 58,40 SAY page
   STORE page+1 TO page
   EJECT
   STORE ep+ "EUD SPECIAL PURCHASED EQUIPMENT" TO mnote
   STORE 6 TO row
   @ 0,20 SAY mnote
   @ 1,55 SAY "DEPRECIATION"
   @ 1,70 SAY "OPERATION &"
   @ 2,0 SAY "ITEM"
   @ 2,13 SAY "NOMENCLATURE"
   @ 2,40 SAY "ACQUISITION"
   @ 2,55 SAY "COST"
```

```
         @ 2,70 SAY "MAINTENANCE"
         @ 3,0 SAY "NO."
         @ 3,13 SAY "   & FSN"
         @ 3,40 SAY "COST"
         @ 3,55 SAY "(Unfunded)"
         @ 3,70 SAY "COST (Funded)&epotf"

      ENDIF #=eudmark

      IF #=stop
         @ 58,40 SAY page
         EJECT
         SET FORMAT TO SCREEN
         STORE 'F' TO mmore
         LOOP
      ENDIF

      STORE eno TO meno2
      STORE str(eno,4,0) TO meno

ENDDO WHILE mmore

SELECT SECOND
USE
*       * deselect temporary
*       * and delete
DELETE FILE teq
DELETE FILE teq.ndx
SELECT PRIMARY
@ 21,0
RETURN
*
*       * END EQRPT.CMD *** RETURN TO EQCOST.CMD
*
```

```
*******************************************************************************
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: GROUPS.CMD
*
*       DESCRIPTION: This program implements the GROUPS definition function
*                    of the "Project Reporting Groups" application begun in
*                    CLUSTERS.CMD.  A group is a set of schedule numbers whereas
*                    a cluster is a set of groups.  This routine lets the user
*                    add, edit, print, or delete groups within a cluster.
*
*
*       CALLS: SCHEDULES.CMD and returns to CLUSTERS.CMD
*
*******************************************************************************
SET COLON OFF
SET TALK OFF

DO WHILE !(cmd)<> "Q"
   ERASE
@ 1, 21 SAY "PROJECT REPORTING GROUPS OF"
@ 3,  2 SAY "-----------------------------------------------------------------
------------"
@ 4,  2 SAY
"|
|"
@ 5,  2 SAY "|  1>                       2>
3>                 |"
@ 6,  2 SAY
"|
|"
@ 7,  2 SAY
"|
|"
@ 8,  2 SAY "|  4>                       5>
6>                 |"
@ 9,  2 SAY
"|
|"
@ 10, 2 SAY
"|
|"
@ 11, 2 SAY "|  7>                       8>
9>                 |"
@ 12, 2 SAY
"|
|"
@ 13, 2 SAY
```

```
"|
|"
@ 14, 2 SAY "| 10>                      11>
12>                     |"
@ 15, 2 SAY
"|
|"
@ 16, 2 SAY
"|============================================================================|
"
@ 17, 2 SAY
"|
|"
@ 18, 2 SAY "|     E to edit/display/add a group        P to print schedule
no.'s        |"
@ 19, 2 SAY "|     D to delete a group                  Q to quit to cluster
menu        |"
@ 20, 2 SAY
"|
|"
@ 21, 2 SAY "| OPTION ?"
@ 21,78 SAY "|"
@ 22, 2 SAY "-----------------------------------------------------------------
------------"
@ 1,50 GET mcname
   CLEAR GETS

   USE groups INDEX groups
   *    * mkey is cluster number
   FIND "&mkey"
   *    * loop to display group names on above spread sheet
   DO WHILE cno=mcno.AND.(.NOT.EOF).AND.#<>0
      IF gno = 7
         STORE 2 TO row
      ELSE
         IF gno > 9
            STORE 3 TO row
         ELSE
            STORE int(gno/4) TO row
         ENDIF gno > 9
      ENDIF gno = 7
      STORE gno-row*3 TO col
      @ row*3+5,25*(col-1)+8 SAY gname
      SKIP
   ENDDO WHILE cno=mcno.AND.(.NOT.EOF).AND.#<>0
   *    * set defaults
   STORE 1 TO mgno
   STORE "E" TO cmd
   STORE "Y" TO mnew
   *
   * main loop for add, edit and delet
   *
```

```
  DO WHILE !(cmd)<> 'Q' .AND.!(cmd)<> 'B'

     @ 21, 13 GET cmd PICTURE "!"
     READ
     CLEAR GETS

     IF !(cmd)= 'E' .OR.!(cmd)= 'D'
        @ 21,18 SAY "which item no.? (1-12)"
        @ 21,41 GET mgno PICTURE "99"
        READ
        CLEAR GETS
        IF mgno > 0 .AND. mgno < 13
           STORE mkey+str(mgno,2,0) TO mkey
           FIND "&mkey"
           IF !(cmd)= 'E'
              IF #=0
                 @ 21,48 SAY "adding new group? (y/n)"
                 @ 21,73 GET mnew
                 READ
                 CLEAR GETS
                 IF !(mnew)= 'Y'
                    APPEND BLANK
                    REPLACE gno WITH mgno, cno WITH mcno
                    IF gno = 7
                       STORE 2 TO row
                    ELSE
                       IF gno > 9
                          STORE 3 TO row
                       ELSE
                          STORE int(gno/4) TO row
                       ENDIF gno > 9
                    ENDIF gno = 7
                    STORE gno-row*3 TO col
STORE "                                    Enter the group name.;
                      " TO mnotel
                    @ 21,18 GET mnotel
                    CLEAR GETS
                    @ row*3+5,25*(col-1)+8 GET gname
                    READ
                    CLEAR GETS
                    STORE gname TO mgname
                    STORE "A" TO cmd
                    ***********
                    DO schedule
                    ***********
                    STORE "B" TO cmd
                 ELSE
                    @ 21,18 SAY
"                                                          "

                 ENDIF !(mnew) = 'Y'
              ELSE
                 STORE gname TO mgname
```

```
            STORE "E" TO cmd
            ************
            DO schedule
            ************
            STORE "B" TO cmd
         ENDIF # = 0
      ELSE
         IF #=0
            STORE "No such Group.  " TO mnote
            @ 21,50 GET mnote
            CLEAR GETS
         ELSE
            @ 21,50 SAY "sure? (y/n)"
            @ 21,62 GET mnew
            READ
            CLEAR GETS
            IF !(mnew)= 'Y'
               IF gno = 7
                  STORE 2 TO row
               ELSE
                  IF gno > 9
                     STORE 3 TO row
                  ELSE
                     STORE int(gno/4) TO row
                  ENDIF gno > 9
               ENDIF gno = 7
               @ 21,18 SAY
"                                                 "
               STORE gno-row*3 TO col
               STORE " WAIT...PROCESSING  " TO mnote4
               @ row*3+5,25*(col-1)+8 GET mnote4
               CLEAR GETS
               USE schedule INDEX schedule
               DELETE FOR cno=mcno.AND.gno=mgno
               PACK
               USE groups INDEX groups
               DELETE FOR cno=mcno.AND.gno=mgno
               PACK
               @ row*3+5,25*(col-1)+8 SAY "                    "
            ELSE
               IF !(mnew)= 'N'
                  @ 21,18 SAY
"                                                    "
               ENDIF !(mnew)= 'N'
            ENDIF !(mnew) = 'Y'
            STORE 'E' TO cmd
            STORE '1' TO gno
        ENDIF # = 0
      ENDIF mgno > 0 .AND. mgno < 13
   ENDIF !(mnew)= 'Y'
ENDIF #=0
```

```
      *         * print routine to list all schedule numbers in group
      IF !(cmd)= 'P'
         @ 21,18 SAY "which item no.? (1-12)"
         @ 21,41 GET mgno PICTURE "99"
         READ
         CLEAR GETS
         IF mgno>0.AND.mgno<13
            STORE mkey+str(mgno,2,0) TO mkey
            FIND "&mkey"
            IF #=0
               STORE "Not a Project Reporting Group" TO mnote5
               @ 21,48 GET mnote5
               CLEAR GETS
            ELSE
               STORE gname TO mgname
               @ 21,18 SAY
"                                                           "
               STORE " WAIT...PREPARING REPORT FOR PRINT" TO mnote5
               @ 21,20 GET mnote5
               CLEAR GETS
               SELECT PRIMARY
               USE schedule INDEX schedule
               FIND '&mkey'
               IF EOF
                  STORE " There are no Schedules Numbers in Group" TO mnote5
                  @ 21,18 GET mnote5
                  @ 21,40 GET mgname
                  CLEAR GETS
               ELSE
                  STORE p.sno TO msno
                  SELECT SECOND
                  USE pdf INDEX pdf
                  @ 21,18 SAY
"                                                                     "
                  STORE " WAIT...PRINTING REPORT " TO mnote5
                  @ 21,20 GET mnote5
                  CLEAR GETS
                  SET FORMAT TO PRINT
DO WHILE .NOT.EOF.AND.(p.cno=mcno).AND.(p.gno=mgno)
@ 2,10 SAY "18th ENG BDE - Automated Construction Management System"
@ 4,5 SAY "Schedule Numbers For Project Reporting Group:"
@ 4,51 SAY mgname
@ 6,5 SAY "Sch. No.  Battalion         Title                    Location"
@ 7,5 SAY "========  =========  ======================    ================= "
STORE 9 TO mrow2
DO WHILE .NOT. EOF.AND.(mrow2<62).AND.(p.cno=mcno).AND.(p.gno=mgno)
      SELECT SECOND
      FIND '&msno'
      @ mrow2,6 SAY s.sno
      @ mrow2,18 SAY s.bn
      @ mrow2,28 SAY s.title
      @ mrow2,54 SAY s.loc
      STORE mrow2+2 TO mrow2
      SELECT PRIMARY
      SKIP +1
      STORE p.sno TO msno
      ENDDO
                     EJECT
                  ENDDO WHILE .NOT.EOF.AND.(p.cno=mcno).AND.(p.g
                  EJECT
                  SET FORMAT TO SCREEN
                  USE groups INDEX groups
                  @ 21,18 SAY
"                                                            "
                  STORE 'E' TO cmd
               ENDIF EOF
            ENDIF #=0
         ENDIF mgno>0.AND.mgno<13
      ENDIF !(cmd)= 'P'

      *         * reset mkey'to cluster
      STORE $(mkey,1,1) TO mkey

   ENDDO WHILE !(cmd)<> 'Q' AND. !(cmd)<> 'B'

ENDDO WHILE !(cmd)<> 'Q'
USE
STORE 'B' TO cmd
ERASE
RETURN
*
*       *  END GROUPS.CMD RETURN TO CLUSTERS.CMD
*
```

```
******************************************************************************
*
*
*                         A - C - M - S  Ver. 84.0
*
*                                 USA-CERL
*
*       PROGRAM: INSTALL.CMD
*
*       DESCRIPTION: Installation program for the ACMS system.  This
*                    produces a memory variable file--INSTALL.MEM--
*                    that contains printer and console control codes.  The
*                    console control is optional and defaults to a space--
*                    ASCII decimal 32.  The printer control codes are
*                    necessary for 80 column printers and because some ACMS
*                    reports require different pitches.  The program also
*                    lets the user enter the title of the organization to
*                    appear in the menu program (ACMS.CMD).
*
*       CALLS: ACMS.CMD
*
******************************************************************************
SET TALK OFF
SET COLON OFF
*       * display system banner
?
?
?
?
?
?
?
?
?
?
?
? '       ___________    __________     ___   ___               __________'
? '      / ______   /   /  ____   /    /   \ /   \             /  ____   /'
? '     / /     /  /   /  /   /__/    /  /\ \/ /\ \           /  /   /__/ '
? '    / /_____/  /   /  /           /  /  \  /  \ \         /  /_______  '
? '   /  _____   /   /  /           /  /    \/    \ \       /_______   /  '
? '  /  /    /  /   /  /    ___    /  /            \ \      ___    /  /   '
? ' /  /    /  /   /  /____/  /   /  /              \ \    /  /___/  /    '
? '/__/    /__/   /__________/   /__/                \_\  /_________/     '
?
?
?
?
?
?
?
```

```
?
?
? '                               Version 84.0'
?
?
?
? '                  AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM'
?
? '                         US ARMY CORPS OF ENGINEERS'
?
? '                Construction Engineering Research Laboratory'
?
?
?
*       * wait
STORE 0 TO lmarktime
STORE 200 TO lmarch
DO WHILE lmarktime<lmarch
   STORE lmarktime+1 TO lmarktime
ENDDO
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
*       * start installation program here
?
? '                THIS IS THE A-C-M-S INSTALLATION PROGRAM.'
?
? 'YOU WILL BE ASKED TO SPECIFY THE CONTROL CODES REQUIRED BY YOUR PRINTER'
? 'AND CONSOLE.  TO DETERMINE THE CONTROL SEQUENCES YOU WILL NEED TO REFER'
? 'TO YOUR SYSTEM MANUALS.'
?
?
? 'THE INSTALLATION PROGRAM WILL PROMPT YOU FOR THE COMMANDS.  THE PROMPTS'
? 'WILL ASK YOU TO ENTER THE NUMBER OF REQUIRED CODE(S) AND THEN THE '
? '***DECIMAL*** VALUE OF THE ASCII CODE(S).'
?
?
? 'NOTE:THE CONSOLE INSTALLATION PART OF THIS PROGRAM IS OPTIONAL.'
? '     IF YOU DO NOT WANT TO USE ANY SPECIAL EFFECTS SUCH AS '
? '     REVERSE VIDEO OR REVERSE BLINK, THEN ANSWER NO TO THE'
? '     FIRST TWO QUESTIONS.  THE DEFAULT CODE 32 IS A SPACE AND'
? '     WILL HAVE NO EFFECT.'
?
?
? 'PRESS <RETURN> TO BEGIN..'
SET CONSOLE OFF
WAIT
SET CONSOLE ON
IF FILE('INSTALL.MEM')
   *    * get old control codes
   RESTORE FROM install ADDITIVE
ELSE
   *    * setup defaults to blank
   *    * start reverse video
   STORE ' ' TO rv
   *    * stop reverse video
   STORE ' ' TO rvoff
   *    * 10 pitch
   STORE ' ' TO p10
   *    * 12 pitch
   STORE ' ' TO p12
   *    * 17 pitch
   STORE ' ' TO p17
   *    * start emphasized print
   STORE ' ' TO ep
   *    * stop emphasized print
   STORE ' ' TO epoff
   *    * start double wide
   STORE ' ' TO dw
   *    * organization title
   STORE '                              ' TO orgname
   *    * bell
   STORE CHR(7) TO bell
ENDIF
*       * get length of strings
STORE LEN(p10)   TO lp10
STORE LEN(p12)   TO lp12
STORE LEN(p17)   TO lp17
STORE LEN(ep)    TO lep
STORE LEN(epoff) TO lepoff
STORE LEN(rv)    TO lrv
STORE LEN(rvoff) TO lrvoff
STORE LEN(dw)    TO ldw
ERASE
STORE 'N' TO lok
*       * loop until console codes are ok
```

```
DO WHILE lok<>'Y'
*
*---------------------get codes for reverse video on--------------------------
*
   STORE STR(lrv,1) TO lmno
   ? '                     CONSOLE INSTALLATION - OPTIONAL'
   @ 3,0 SAY 'CURRENT COMMAND FOR START REVERSE VIDEO (OR REVERSE BLINK)'
   @ 4,0 SAY 'REQUIRES &lmno CODE(S) AS FOLLOWS:'
   STORE 1 TO lno
   STORE '1' TO lnumber
   STORE 6 TO lrow
      DO WHILE lno<=lrv
         STORE $(rv,lno,1) TO lrv&lnumber
         STORE RANK(lrv&lnumber) TO lcode
         @ lrow,10 SAY 'CODE.. &lnumber .. IS.... ' GET lcode PICTURE '99'
         CLEAR GETS
         STORE lno+1 TO lno
         STORE STR(&lnumber+1,1) TO lnumber
         STORE lrow+1 TO lrow
      ENDDO
      STORE 'N' TO lchange
      STORE lrow+1 TO lrow
      @ lrow,0 SAY 'DO YOU WANT TO CHANGE THESE CODES? ' GET lchange PICTURE
'!'
      READ
      IF lchange='Y'
        RELEASE ALL LIKE lrv?
        STORE 0 TO lrv
        STORE lrow+2 TO lrow
        STORE '1' TO lnumber
        STORE  1  TO lno
        @ lrow,2 SAY 'ENTER THE NUMBER OF CODES: ' GET lrv PICTURE '9'
        READ
        DO WHILE lno<=lrv
          STORE lrow+1 TO lrow
          STORE 0 TO lrv&lnumber
          @ lrow,9 SAY 'ENTER DECIMAL CODE: ' GET lrv&lnumber PICTURE '99'
          READ
          STORE lno+1 TO lno
          STORE STR(&lnumber+1,1) TO lnumber
        ENDDO
        STORE CHR(lrv1) TO rv
        IF lrv>1
          STORE 2 to lno
          STORE '2' TO lnumber
          STORE CHR(lrv1) TO rv
          DO WHILE lno<=lrv
             STORE rv+CHR(lrv&lnumber) TO rv
             STORE lno+1 TO lno
             STORE STR(&lnumber+1,1) TO lnumber
          ENDDO
        ENDIF
```

```
     ENDIF
     ERASE
*
*---------------------get codes for reverse video off-------------------------
*
     STORE STR(lrvoff,1) TO lmno
     ? '                     CONSOLE INSTALLATION - OPTIONAL'
     @ 3,0 SAY 'CURRENT COMMAND FOR STOP REVERSE VIDEO (OR REVERSE BLINK)'
     @ 4,0 SAY 'REQUIRES &lmno CODE(S) AS FOLLOWS:'
     STORE 1 TO lno
     STORE '1' TO lnumber
     STORE 6 TO lrow
     DO WHILE lno<=lrvoff
        STORE $(rvoff,lno,1) TO lrvoff&lnumber
        STORE RANK(lrvoff&lnumber) TO lcode
        @ lrow,10 SAY 'CODE.. &lnumber .. IS.... ' GET lcode PICTURE '99'
        CLEAR GETS
        STORE lno+1 TO lno
        STORE STR(&lnumber+1,1) TO lnumber
        STORE lrow+1 TO lrow
     ENDDO
     STORE 'N' TO lchange
     STORE lrow+1 TO lrow
     @ lrow,0 SAY 'DO YOU WANT TO CHANGE THESE CODES? ' GET lchange PICTURE
'!'
     READ
     IF lchange='Y'
        RELEASE ALL LIKE lrvoff?
        STORE 0 TO lrvoff
        STORE lrow+2 TO lrow
        STORE '1' TO lnumber
        STORE  1  TO lno
        @ lrow,2 SAY 'ENTER THE NUMBER OF CODES: ' GET lrvoff PICTURE '9'
        READ
        DO WHILE lno<=lrvoff
           STORE lrow+1 TO lrow
           STORE 0 TO lrvoff&lnumber
           @ lrow,9 SAY 'ENTER DECIMAL CODE: ' GET lrvoff&lnumber PICTURE '99'
           READ
           STORE lno+1 TO lno
           STORE STR(&lnumber+1,1) TO lnumber
        ENDDO
        STORE CHR(lrvoff1) TO rvoff
        IF lrvoff>1
          STORE 2 to lno
          STORE '2' TO lnumber
          DO WHILE lno<=lrvoff
             STORE rvoff+CHR(lrvoff&lnumber) TO rvoff
             STORE lno+1 TO lno
             STORE STR(&lnumber+1,1) TO lnumber
          ENDDO
       ENDIF
```

```
   ENDIF
*
*---------------------test reverse video---------------------------------------
*
   ERASE
   STORE 'N' TO ltest
   ? '                    CONSOLE INSTALLATION - OPTIONAL'
   @ 3,0 SAY 'DO YOU WANT TO TEST YOUR REVERSE VIDEO (REVERSE BLINK) COMAND?
';
   GET ltest PICTURE '!'
   READ
   IF ltest='Y'
      @ 10,18 SAY rv+'***********************'+rvoff
      @ 12,19 SAY 'IS THIS WHAT YOU WANT? 'GET lok PICTURE '!'
      READ
   ELSE
      STORE 'Y' TO lok
   ENDIF
   ERASE
ENDDO
*----------------------printer controls----------------------------------------
*
STORE 'N' TO lok
*       * loop until printer codes are ok
DO WHILE lok<>'Y'
*----------------------10 characters per inch----------------------------------
*
   ERASE
   STORE STR(lp10,1) TO lmno
   ? '                         PRINTER INSTALLATION'
   @ 3,0 SAY 'CURRENT COMMAND FOR 10 CHARACTERS PER INCH'
   @ 4,0 SAY 'REQUIRES &lmno CODE(S) AS FOLLOWS:'
   STORE 1 TO lno
   STORE '1' TO lnumber
   STORE 6 TO lrow
   DO WHILE lno<=lp10
      STORE $(p10,lno,1) TO lp10&lnumber
      STORE RANK(lp10&lnumber) TO lcode
      @ lrow,10 SAY 'CODE.. &lnumber .. IS.... ' GET lcode PICTURE '99'
      CLEAR GETS
      STORE lno+1 TO lno
      STORE STR(&lnumber+1,1) TO lnumber
      STORE lrow+1 TO lrow
   ENDDO
   STORE 'N' TO lchange
   STORE lrow+1 TO lrow
   @ lrow,0 SAY 'DO YOU WANT TO CHANGE THESE CODES? ' GET lchange PICTURE '!'
   READ
   IF lchange='Y'
      RELEASE ALL LIKE lp10?
      STORE 0 TO lp10
      STORE lrow+2 TO lrow
      STORE '1' TO lnumber
      STORE  1  TO lno
      @ lrow,2 SAY 'ENTER THE NUMBER OF CODES: ' GET lp10 PICTURE '9'
      READ
      DO WHILE lno<=lp10
         STORE lrow+1 TO lrow
         STORE 0 TO lp10&lnumber
         @ lrow,9 SAY 'ENTER DECIMAL CODE: ' GET lp10&lnumber PICTURE '99'
         READ
         STORE lno+1 TO lno
         STORE STR(&lnumber+1,1) TO lnumber
       ENDDO
       STORE CHR(lp101) TO p10
       IF lp10>1
          STORE 2 to lno
          STORE '2' TO lnumber
          DO WHILE lno<=lrvoff
             STORE p10+CHR(lp10&lnumber) TO p10
             STORE lno+1 TO lno
             STORE STR(&lnumber+1,1) TO lnumber
          ENDDO
       ENDIF
    ENDIF
*
*---------------------12 characters per inch---------------------------------
*
    ERASE
    STORE STR(lp12,1) TO lmno
    ? '                       PRINTER INSTALLATION'
    @ 3,0 SAY 'CURRENT COMMAND FOR 12 CHARACTERS PER INCH'
    @ 4,0 SAY 'REQUIRES &lmno CODE(S) AS FOLLOWS:'
    STORE 1 TO lno
    STORE '1' TO lnumber
    STORE 6 TO lrow
    DO WHILE lno<=lp12
       STORE $(p12,lno,1) TO lp12&lnumber
       STORE RANK(lp12&lnumber) TO lcode
       @ lrow,10 SAY 'CODE.. &lnumber .. IS.... ' GET lcode PICTURE '99'
       CLEAR GETS
       STORE lno+1 TO lno
       STORE STR(&lnumber+1,1) TO lnumber
       STORE lrow+1 TO lrow
    ENDDO
    STORE 'N' TO lchange
    STORE lrow+1 TO lrow
    @ lrow,0 SAY 'DO YOU WANT TO CHANGE THESE CODES? ' GET lchange PICTURE '!'
    READ
    IF lchange='Y'
       RELEASE ALL LIKE lp12?
       STORE 0 TO lp12
       STORE lrow+2 TO lrow
       STORE '1' TO lnumber
       STORE  1  TO lno
       @ lrow,2 SAY 'ENTER THE NUMBER OF CODES: ' GET lpl2 PICTURE '9'
       READ
       DO WHILE lno<=lpl2
          STORE lrow+1 TO lrow
          STORE 0 TO lpl2&lnumber
          @ lrow,9 SAY 'ENTER DECIMAL CODE: ' GET lpl2&lnumber PICTURE '99'
          READ
          STORE lno+1 TO lno
          STORE STR(&lnumber+1,1) TO lnumber
       ENDDO
       STORE CHR(lpl21) TO pl2
       IF lpl2>1
          STORE 2 to lno
          STORE '2' TO lnumber
          DO WHILE lno<=lpl2
             STORE pl2+CHR(lpl2&lnumber) TO pl2
             STORE lno+1 TO lno
             STORE STR(&lnumber+1,1) TO lnumber
          ENDDO
       ENDIF
    ENDIF
*
*----------------------17 characters per inch---------------------------------
*
    ERASE
    STORE STR(lpl7,1) TO lmno
    ? '                          PRINTER INSTALLATION'
    @ 3,0 SAY 'CURRENT COMMAND FOR 17 CHARACTERS PER INCH'
    @ 4,0 SAY 'REQUIRES &lmno CODE(S) AS FOLLOWS:'
    STORE 1 TO lno
    STORE '1' TO lnumber
    STORE 6 TO lrow
    DO WHILE lno<=lpl7
       STORE $(pl7,lno,1) TO lpl7&lnumber
       STORE RANK(lpl7&lnumber) TO lcode
       @ lrow,10 SAY 'CODE.. &lnumber .. IS.... ' GET lcode PICTURE '99'
       CLEAR GETS
       STORE lno+1 TO lno
       STORE STR(&lnumber+1,1) TO lnumber
       STORE lrow+1 TO lrow
    ENDDO
    STORE 'N' TO lchange
    STORE lrow+1 TO lrow
    @ lrow,0 SAY 'DO YOU WANT TO CHANGE THESE CODES? ' GET lchange PICTURE '!'
    READ
    IF lchange='Y'
       RELEASE ALL LIKE lpl7?
       STORE 0 TO lpl7
       STORE lrow+2 TO lrow
       STORE '1' TO lnumber
       STORE  1  TO lno
       @ lrow,2 SAY 'ENTER THE NUMBER OF CODES: ' GET lpl7 PICTURE '9'
       READ
       DO WHILE lno<=lpl7
          STORE lrow+1 TO lrow
          STORE 0 TO lpl7&lnumber
          @ lrow,9 SAY 'ENTER DECIMAL CODE: ' GET lpl7&lnumber PICTURE '99'
          READ
          STORE lno+1 TO lno
          STORE STR(&lnumber+1,1) TO lnumber
       ENDDO
       STORE CHR(lpl71) TO pl7
       IF lpl7>1
          STORE 2 to lno
          STORE '2' TO lnumber
          DO WHILE lno<=lpl7
             STORE pl7+CHR(lpl7&lnumber) TO pl7
             STORE lno+1 TO lno
             STORE STR(&lnumber+1,1) TO lnumber
          ENDDO
       ENDIF
    ENDIF
*
*----------------------start emphasized print---------------------------------
*
    ERASE
    STORE STR(lep,1) TO lmno
    ? '                       PRINTER INSTALLATION'
    @ 3,0 SAY 'CURRENT COMMAND FOR START EMPHASIZED PRINTING'
    @ 4,0 SAY 'REQUIRES &lmno CODE(S) AS FOLLOWS:'
    STORE 1 TO lno
    STORE '1' TO lnumber
    STORE 6 TO lrow
    DO WHILE lno<=lep
       STORE $(ep,lno,1) TO lep&lnumber
       STORE RANK(lep&lnumber) TO lcode
       @ lrow,10 SAY 'CODE.. &lnumber .. IS.... ' GET lcode PICTURE '99'
       CLEAR GETS
       STORE lno+1 TO lno
       STORE STR(&lnumber+1,1) TO lnumber
       STORE lrow+1 TO lrow
    ENDDO
    STORE 'N' TO lchange
    STORE lrow+1 TO lrow
    @ lrow,0 SAY 'DO YOU WANT TO CHANGE THESE CODES? ' GET lchange PICTURE '!'
    READ
    IF lchange='Y'
       RELEASE ALL LIKE lep?
       STORE 0 TO lep
       STORE lrow+2 TO lrow
       STORE '1' TO lnumber
       STORE  1  TO lno
       @ lrow,2 SAY 'ENTER THE NUMBER OF CODES: ' GET lep PICTURE '9'
       READ
       DO WHILE lno<=lep
          STORE lrow+1 TO lrow
          STORE 0 TO lep&lnumber
          @ lrow,9 SAY 'ENTER DECIMAL CODE: ' GET lep&lnumber PICTURE '99'
          READ
          STORE lno+1 TO lno
          STORE STR(&lnumber+1,1) TO lnumber
       ENDDO
       STORE CHR(lep1) TO ep
       IF lep>1
          STORE 2 to lno
          STORE '2' TO lnumber
          DO WHILE lno<=lep
             STORE ep+CHR(lep&lnumber) TO ep
             STORE lno+1 TO lno
             STORE STR(&lnumber+1,1) TO lnumber
          ENDDO
       ENDIF
    ENDIF
*
*-----------------------stop emphasized print--------------------------------
*
    ERASE
    STORE STR(lepoff,1) TO lmno
    ? '                       PRINTER INSTALLATION'
    @ 3,0 SAY 'CURRENT COMMAND FOR STOP EMPASIZED PRINTING'
    @ 4,0 SAY 'REQUIRES &lmno CODE(S) AS FOLLOWS:'
    STORE 1 TO lno
    STORE '1' TO lnumber
    STORE 6 TO lrow
    DO WHILE lno<=lepoff
       STORE $(epoff,lno,1) TO lepoff&lnumber
       STORE RANK(lepoff&lnumber) TO lcode
       @ lrow,10 SAY 'CODE.. &lnumber .. IS.... ' GET lcode PICTURE '99'
       CLEAR GETS
       STORE lno+1 TO lno
       STORE STR(&lnumber+1,1) TO lnumber
       STORE lrow+1 TO lrow
    ENDDO
    STORE 'N' TO lchange
    STORE lrow+1 TO lrow
    @ lrow,0 SAY 'DO YOU WANT TO CHANGE THESE CODES? ' GET lchange PICTURE '!'
    READ
    IF lchange='Y'
       RELEASE ALL LIKE lepoff?
       STORE 0 TO lepoff
       STORE lrow+2 TO lrow
       STORE '1' TO lnumber
       STORE  1  TO lno
       @ lrow,2 SAY 'ENTER THE NUMBER OF CODES: ' GET lepoff PICTURE '9'
       READ
       DO WHILE lno<=lepoff
          STORE lrow+1 TO lrow
          STORE 0 TO lepoff&lnumber
          @ lrow,9 SAY 'ENTER DECIMAL CODE: ' GET lepoff&lnumber PICTURE '99'
          READ
          STORE lno+1 TO lno
          STORE STR(&lnumber+1,1) TO lnumber
       ENDDO
       STORE CHR(lepoff1) TO epoff
       IF lepoff>1
          STORE 2 to lno
          STORE '2' TO lnumber
          DO WHILE lno<=lepoff
             STORE epoff+CHR(lepoff&lnumber) TO epoff
             STORE lno+1 TO lno
             STORE STR(&lnumber+1,1) TO lnumber
          ENDDO
       ENDIF
    ENDIF
*
*-----------------------start double wide print--------------------------------
*
   ERASE
   STORE STR(ldw,1) TO lmno
   ? '                          PRINTER INSTALLATION'
   @ 3,0 SAY 'CURRENT COMMAND FOR START DOUBLE WIDE PRINTING'
   @ 4,0 SAY 'REQUIRES &lmno CODE(S) AS FOLLOWS:'
   STORE 1 TO lno
   STORE '1' TO lnumber
   STORE 6 TO lrow
   DO WHILE lno<=ldw
      STORE $(dw,lno,1) TO ldw&lnumber
      STORE RANK(ldw&lnumber) TO lcode
      @ lrow,10 SAY 'CODE.. &lnumber .. IS.... ' GET lcode PICTURE '99'
      CLEAR GETS
      STORE lno+1 TO lno
      STORE STR(&lnumber+1,1) TO lnumber
      STORE lrow+1 TO lrow
    ENDDO
    STORE 'N' TO lchange
    STORE lrow+1 TO lrow
    @ lrow,0 SAY 'DO YOU WANT TO CHANGE THESE CODES? ' GET lchange PICTURE '!'
    READ
    IF lchange='Y'
       RELEASE ALL LIKE ldw?
       STORE 0 TO ldw
       STORE lrow+2 TO lrow
       STORE '1' TO lnumber
       STORE  1  TO lno
       @ lrow,2 SAY 'ENTER THE NUMBER OF CODES: ' GET ldw PICTURE '9'
       READ
       DO WHILE lno<=ldw
```

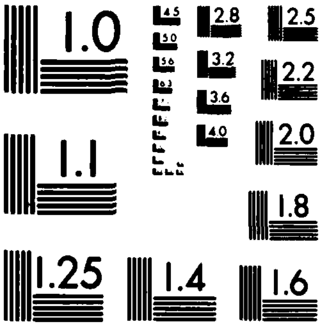
```
          STORE lrow+1 TO lrow
          STORE 0 TO ldw&lnumber
          @ lrow,9 SAY 'ENTER DECIMAL CODE: ' GET ldw&lnumber PICTURE '99'
          READ
          STORE lno+1 TO lno
          STORE STR(&lnumber+1,1) TO lnumber
       ENDDO
       STORE CHR(ldw1) TO dw
       IF ldw>1
          STORE 2 to lno
          STORE '2' TO lnumber
          DO WHILE lno<=ldw
             STORE dw+CHR(ldw&lnumber) TO dw
             STORE lno+1 TO lno
             STORE STR(&lnumber+1,1) TO lnumber
          ENDDO
       ENDIF
    ENDIF
*
*----------------------test printer installation------------------------------
*
    ERASE
    STORE 'Y' TO ltest
    ? '                   PRINTER INSTALLATION'
    @ 3,0 SAY 'DO YOU WANT TO TEST YOUR PRINTER INSTALLATION? ';
    GET ltest PICTURE '!'
    READ
    IF ltest='Y'
       SET FORMAT TO PRINT
       @ 1,0 SAY p10+'This is ten characters per inch.'
       @ 3,0 SAY p12+'This is twelve characters per inch.'
       @ 5,0 SAY p17+'This is seventeen characters per inch.'
       @ 7,0 SAY ep+p10+'This is ten characters per inch;
 emphasized.'+epoff
       @ 9,0 SAY dw+'This is DOUBLE WIDE 10 pitch.'+p10
       EJECT
       SET FORMAT TO SCREEN
    ELSE
       STORE 'Y' TO lok
       LOOP
    ENDIF
    @ 5,0 SAY 'IS THE PRINT OUT CORRECT? ' GET lok PICTURE '!'
    READ
    ERASE
ENDDO
*
*----------------------organizational title-----------------------------------
*
ERASE
? '                   TITLE TO APPEAR IN MENUS'
@ 3,0 SAY 'CURRENT ORGANIZATION TITLE IS: ' GET orgname
CLEAR GETS
```

```
STORE 'Y' TO lchange
@ 5,0 SAY 'DO YOU WANT TO CHANGE THE TITLE? ' GET lchange PICTURE '!'
READ
IF lchange='Y'
   @ 5,0
   @ 5,0 SAY 'ENTER YOUR ORGANIZATION TITLE: ' GET orgname
   @ 6,0 SAY '   (30 Characters maximum)'
   READ
   STORE TRIM(orgname) TO orgname
   STORE LEN(orgname) TO lorg
   DO WHILE @(' ',orgname)=1
      STORE lorg-1 TO lorg
      STORE $(orgname,2,lorg) TO orgname
   ENDDO
   STORE LEN(orgname) TO lorg
   STORE 30 TO lmax
   DO WHILE lorg<lmax
      STORE ' '+orgname TO orgname
      STORE lorg+1 TO lorg
      IF lorg<lmax
         STORE orgname+' ' TO orgname
         STORE lorg+1 TO lorg
      ENDIF
   ENDDO
ENDIF
*
*-------------------installation complete---------------------------------
*       * release temporary memory variables
ERASE
? 'WAIT.. SAVING CONTROL CODES TO DISK'
RELEASE ALL LIKE l*
SAVE TO install
ERASE
?
?
?
?
?
?
?
?
? 'THE INSTALLATION IS COMPLETE.  YOUR CONTROL CODES HAVE BEEN SAVED'
? 'TO DISK IN A FILE NAMED - INSTALL.MEM.  THIS FILE MUST BE ON THE '
? 'DISK WITH THE ACMS dBASE-II COMMAND FILES.  YOU MAY CHANGE THESE'
? 'CODES AT ANY TIME BY RUNNING THE INSTALL PROGRAM AGAIN.'
?
?
?
?
?
?
?
```

```
?
?
?
? 'PRESS <RETURN> TO BEGIN ACMS..'
SET CONSOLE OFF
WAIT
SET CONSOLE ON
*
*******
DO ACMS
*******
*
QUIT
*
* END INSTALL.CMD
*
```

AD-A143 032 AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM (ACMS) VOLUME 2/3
2 PROGRAM DOCUMENTATION(U) CONSTRUCTION ENGINEERING
RESEARCH LAB (ARMY) CHAMPAIGN IL C E HERRING JUN 84

UNCLASSIFIED CERL-TR-P-158-VOL-2 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
********************************************************************************
*
*                        A - C - M - S  Ver. 84.0
*
*                                USA-CERL
*
*       PROGRAM: IWPR.CMD
*
*       DESCRIPTION: Implements initial WPR activity data base.  An IWPR is a
*                    special case of a WPR in which the subtasks that make
*                    up the task (cpm i-j's) are listed.  The data on the
*                    IWPR consist of i, j, description, and scheduled
*                    manhours.  These uata are used to calculate the percent
*                    complete of the task.  This program provides for input,
*                    change, deletion, and printout of the IWPR data.  Total
*                    scheduled and actual manhours, and actual percent complete
*                    are stored in PS.DBF.  These are corrected after changes
*                    and deletions.
*
*       CALLS: none
*
********************************************************************************
ERASE
SET COLON OFF
SET TALK OFF
SET DELETE ON
STORE '1' TO mcmd
SELECT PRIMARY
USE act INDEX act
*       * index is on sno+STR(i,4,0)+STR(j,4,0)
SELECT SECONDARY
USE pdf INDEX pdf
DO WHILE mcmd<> '0'
   STORE '1' TO mcmd
   STORE 'A' TO prn
   *    * default is 'input IWPR data'
   STORE 'XXXXXX' TO msno
   ERASE
   @  2,10 SAY "*****************************************************"
   @  3,10 SAY "*              INITIAL  WPR  DATA  BASE             *"
   @  4,10 SAY "*****************************************************"
   @  5,10 SAY "*                                                   *"
   @  6,10 SAY "*  <1> Input  IWPR data                             *"
   @  7,10 SAY "*                                                   *"
   @  8,10 SAY "*  <2> Change IWPR data                             *"
   @  9,10 SAY "*                                                   *"
   @ 10,10 SAY "*  <3> Delete IWPR data                             *"
   @ 11,10 SAY "*                                                   *"
   @ 12,10 SAY "*  <4> Display/Print IWPR data for a task           *"
   @ 13,10 SAY "*                                                   *"
   @ 14,10 SAY "*  <5> Display/Print tasks that have IWPR data      *"
   @ 15,10 SAY "*                                                   *"
   @ 16,10 SAY "*  <6> Display/Print tasks that have NO IWPR data   *"
   @ 17,10 SAY "*                                                   *"
@ 18,10 SAY "*  <0> Exit                                           *"
@ 19,10 SAY "*                                                     *"
@ 20,10 SAY "*******************************************************"
@ 21,10
@ 22,10 SAY "OPTION? " CET mcmd PICTURE '9'
READ NOUP
CLEAR GETS
IF (mcmd >='1').AND.(mcmd<='4')
   @ 22,10
   @ 22,10 SAY 'Enter Schedule No. : ' GET msno
   READ NOUP
   CLEAR GETS
   SELECT SECONDARY
   FIND '&msno'
   IF #=0
   STORE 'Schedule NOT in Task Data Base' TO note
   @ 22,10
   @ 22,10 GET NOTE
   CLEAR GETS
   STORE 0 TO mark
   DO WHILE mark<75
     STORE mark+1 TO mark
   ENDD
   @ 22,10
   LOOP
   ENDIF

   SELECT PRIMARY
   FIND "&msno"
   IF #=0.AND.mcmd<>'1'
     STORE 'There is no IWPR data for this task' TO note
     @ 22,10  GET note
     CLEAR GETS
     STORE 1 TO nu
     DO WHILE nu<100
        STORE nu+1 TO nu
     ENDDO
     LOOP
   ENDIF


   *          * task in data base -- show user header information
   SELECT SECONDARY
   ERASE
   STORE 'N' to mok
   @ 1,22 SAY "INITIAL WPR DATA BASE"
   @ 3,5  SAY "SCHEDULE NUMBER: " get msno
   @ 3,40 SAY "Title     : " get title
   @ 4,5  SAY "Battalion      : " get bn
   @ 4,40 SAY "Location  : " get loc
   CLEAR GETS
   @ 6, 15 SAY "Is this the correct task? (y/n)" ;
```

```
      GET mok PICTURE '!'
      READ NOUP
      CLEAR GETS
      IF !(mok)<> 'Y'
         LOOP
      ENDIF

   ENDIF
*
*--------------special case of mcmd='3' : option to delete all------------
*
   IF mcmd='3'
      * delete i-j's or all i-j's for task
      @ 1,0
      @ 1,16 SAY "INITIAL WPR DATA BASE -- DELETE I-J's"
      @ 6,0
      STORE 'N' TO mdel
      @ 6,10 SAY "DO YOU WANT TO DELETE ALL I-J's ?" GET mdel PICTURE '!'
      READ NOUP
      IF mdel='Y'
         STORE 'N' TO mdel
         @ 8,18 SAY "ARE YOU SURE ?" GET mdel PICTURE '!'
         READ NOUP
         IF mdel='Y'
             @ 6,0
             ?
             ?
             ? "                             OKAY!"
             SELECT PRIMARY
             DELETE ALL FOR sno=msno
             *  * zero out totals in ps.dbf
             SELECT SECONDARY
             USE ps INDEX ps
             FIND '&msno'
             REPLACE s.smh WITH 0,s.amh WITH 0,s.per WITH 0 NOUPDATE
             USE pdf INDEX pdf
             SELECT PRIMARY
             STORE '1' TO mcmd
             LOOP
          ENDIF
      ENDIF
   ENDIF
*
*-------------------start main case structure-----------------------------
*
   DO CASE

   *    * input mode

      CASE mcmd= '1'
```

```
        *       * variable to store total change in schedule manhours
        STORE 0 TO chg:smh
@ 1,0
@ 1,16 SAY "INITIAL WPR DATA BASE -- INPUT I-J's"
@ 6,0
@ 9,1 SAY  "+--------+--------+-----------------+------------+------------+----------+"
@ 10,1 SAY "| Start  |  End   |   Description   | Man Hours  | Man Hours  | Percent  |"
@ 11,1 SAY "| Node I | Node J |                 | Scheduled  | Actual     | Complete |"
@ 12,1 SAY "+--------+--------+-----------------+------------+------------+----------+"
         SELECT PRIMARY
         STORE "Y" TO mmore
         DO WHILE !(mmore)<>'N'
               STORE 'Y' TO mmore
               STORE "want to add new i-j's ?(y/n)" TO note
               @ 21,5
               @ 21,5 GET note
               CLEAR GETS
               @ 21,34 GET mmore
               READ NOUP
               @ 21,5
               CLEAR GETS
               IF !(mmore)<>"Y"
                  STORE 'N' TO mmore
                  LOOP
               ENDIF
@ 13, 1 SAY "|        |        |                 |            |            |          |"
@ 14, 1 SAY "+--------+--------+-----------------+------------+------------+----------+"
@ 15, 1 SAY "|        |        |                 |            |            |          |"
@ 16, 1 SAY "+--------+--------+-----------------+------------+------------+----------+"
@ 17, 1 SAY "|        |        |                 |            |            |          |"
@ 18, 1 SAY "+--------+--------+-----------------+------------+------------+----------+"
@ 19, 1 SAY "|        |        |                 |            |            |          |"
@ 20, 1 SAY "+--------+--------+-----------------+------------+------------+----------+"
@ 22, 0 SAY "Type CTR-Q: to quit     ; CTR-F: next field       ; CTR-A: previous field ;"
@ 23, 0 SAY "     CTR-W: next screen ; CTR-D: next character   ; CTR-S: next character ;"
            STORE 1 TO nu
            STORE 13 TO row
            DO WHILE (nu<5).AND.!(mmore)<>'N'
               APPEND BLANK
               REPLACE sno WITH msno
               @ row,3  GET i
               @ row,12 GET j
               @ row,21 GET des
               @ row,39 GET smh
               *       * user cannot enter actual manhours
               *       * or percent complete here
               STORE nu+1 TO nu
               STORE row+2 TO row
               READ
               *       * end input condition
               IF (p.i=0).AND.(p.j=0).AND.(p.smh=0)
                 DELETE
                 STORE 5 TO nu
                 LOOP
               ELSE
               *       * accumulate total schedule manhours
               STORE p.smh+chg:smh TO chg:smh
               ENDIF
          ENDDO
       ENDDO WHILE !(mmore)<> 'N'
      *         * record changes in ps.dbf scheduled
      @ 21,0
      STORE "Wait.." to note
      @ 21,5 GET note
      CLEAR GETS
      SELECT SECOND
      USE ps INDEX ps
      FIND '&msno'
      STORE s.smh TO old:smh
      REPLACE s.smh WITH s.smh+chg:smh NOUPDATE
      STORE s.actper TO old:per
      * * must recalculate actual percent complete
      REPLACE s.actper WITH (old:per*old:smh)/s.smh NOUP
      SELECT SECONDARY
      USE pdf INDEX pdf
      RELEASE chg:smh,old:smh,old:per
      STORE 'Y' TO mprn
      @ 21,0
      @ 21,5 say 'Do you want a printout for this Task? ' ;
      GET mprn PICTURE '!'
      READ NOUP
      IF !(mprn)='Y'
        STORE '4' TO mcmd
        STORE 'B' TO prn
      ENDIF
      RELEASE note,mprn


      CASE mcmd='2'

         *      * change IWPR data
         *      * variables to store changes to
         *      * manhours and percent complete
         STORE 0 TO chg:amh,chg:smh,old:per,new:per
@ 1,0
@ 1,15 SAY "INITIAL WPR DATA BASE -- CHANGE I-J's"
@ 6,0 say '                                                        '
STORE 'NOTE: Enter 0 for Start Node to STOP' TO note
@ 7,10 GET note
CLEAR GETS
@ 9,1 SAY  "+--------+--------+-----------------+------------+------------+----------+"
@ 10,1 SAY "| Start  |  End   |   Description   | Man Hours  | Man Hours  | Percent  |"
@ 11,1 SAY "| Node I | Node J |                 | Scheduled  | Actual     | Complete |"
@ 12,1 SAY "+--------+--------+-----------------+------------+------------+----------+"
```

```
          SELECT PRIMARY
          STORE "Y" TO mmore

          DO WHILE mmore<>'N'

@ 13, 1 SAY "|        |        |                 |           |            |          |"
@ 14, 1 SAY "+--------+--------+-----------------+-----------+------------+----------+"
@ 15, 1 SAY "|        |        |                 |           |            |          |"
@ 16, 1 SAY "+--------+--------+-----------------+-----------+------------+----------+"
@ 17, 1 SAY "|        |        |                 |           |            |          |"
@ 18, 1 SAY "+--------+--------+-----------------+-----------+------------+----------+"
@ 19, 1 SAY "|        |        |                 |           |            |          |"
@ 20, 1 SAY "+--------+--------+-----------------+-----------+------------+----------+"
@ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field  ;"
@ 23, 0 SAY "     CTR-W: next screen ;  CTR-D: next character  ;  CTR-S: next character ;"
               STORE 1 TO nu
               STORE 13 TO row
               DO WHILE (nu<5).AND.!(mmore)<>'N'
               STORE 0 TO mi
               @ row, 3 GET mi PICTURE '9999'
               READ NOUP
               CLEAR GETS
               IF mi=0
                  STORE 5 TO nu
                  LOOP
               ENDIF
               STORE 0 TO mj
               @ row,12 GET mj PICTURE '9999'
               READ NOUP
               CLEAR GETS
               STORE msno+str(mi,4,0)+str(mj,4,0) to mkey
               FIND '&mkey'
               IF #=0
                  STORE 'I-J NOT FOUND!'+CHR(7) TO note
                  @ 21,5
                  @ 21,5 GET note
                  CLEAR GETS
                  LOOP
               ENDIF
               STORE p.smh TO tsmh
               STORE p.amh TO tamh
               STORE smh*per+old:per TO old:per
               @ row,21 GET des
               @ row,39 GET smh
               @ row,52 GET amh
               @ row,65 GET per
               STORE nu+1 TO nu
               STORE row+2 TO row
               READ NOUP
               *        * calculate changes
               *        * to scheduled, actual, and percent complete
               STORE p.smh-tsmh+chg:smh TO chg:smh
               STORE p.amh-tamh+chg:amh TO chg:amh
               STORE smh*per+new:per TO new:per
               @ 21,0
            ENDDO

            STORE 'Want to make more changes? ' TO note
            @ 21,5 GET note
            CLEAR GETS
            @ 21,31 GET mmore PICTURE '!'
            READ NOUP
            CLEAR GETS
            @ 21,0
         ENDDO WHILE !(mmore)<> 'N'

         *      * calculate and record changes in ps.dbf
         @ 21,0
         STORE "Wait.."to note
         @ 21,5 GET note
         CLEAR GETS
         SELECT SECOND
         USE ps INDEX ps
         FIND '&msno'
         STORE s.smh TO old:smh
         REPLACE s.smh WITH s.smh+chg:smh,s.amh WITH s.amh+chg:amh NOUP
         STORE s.smh TO new:smh
         STORE s.actper TO temp
         REPLACE s.actper WITH ((old:smh*temp)-old:per+new:per)/new:smh NOUP
         RELEASE chg:smh,chg:amh,new:per,old:per,old:smh,temp,new:smh
         SELECT SECOND
         USE pdf INDEX pdf
      STORE 'Y' TO mprn
      @ 21,0
      @ 21,5 say 'Do you want a printout for this Task? ' ;
      GET mprn PICTURE '!'
      READ NOUP
      IF !(mprn)='Y'
        STORE '4' TO mcmd
        STORE 'B' TO prn
      ENDIF
      RELEASE note,mprn



    CASE mcmd='3'

        *       * variables to store changes
        STORE 0 TO chg:smh,chg:amh,chg:per
@ 6,0
@ 7,0
@ 8,0
STORE 'Enter 0 for Start Node to STOP' to note
```

```
@ 7,10 GET note
CLEAR GETS
@ 9,1 SAY  "+--------+--------+------------------+------------+------------+----------+"
@ 10,1 SAY "| Start  |  End   |   Description    | Man Hours  | Man Hours  | Percent  |"
@ 11,1 SAY "| Node I | Node J |                  | Scheduled  | Actual     | Complete |"
@ 12,1 SAY "+--------+--------+------------------+------------+------------+----------+"
        SELECT PRIMARY
        STORE "Y" TO mmore

        DO WHILE mmore<>'N'

@ 13, 1 SAY "|        |        |                  |            |            |          |"
@ 14, 1 SAY "+--------+--------+------------------+------------+------------+----------+"
           STORE 0 TO mi
           @ 13,3 GET mi PICTURE '9999'
           READ NOUP
           CLEAR GETS
           IF mi=0
              STORE 'N' TO mmore
              LOOP
           ENDIF
           STORE 0 TO mj
           @ 13,12 GET mj PICTURE '9999'
           READ NOUP
           CLEAR GETS
           STORE msno+str(mi,4,0)+str(mj,4,0) to mkey
           FIND '&mkey'
           IF #=0
                 STORE 'I-J NOT FOUND!'+CHR(7) TO notel
                 @ 21,5
                 @ 21,5 GET notel
                 CLEAR GETS
                 LOOP
           ENDIF
          @ 13,21 GET des
          @ 13,39 GET smh
          @ 13,52 GET amh
          @ 13,65 GET per
          CLEAR GETS
          STORE 'N' TO mdel
          STORE "DO YOU WANT TO DELETE THIS I-J ? " TO note
          @ 21,5 GET note
          CLEAR GETS
          @ 21,40 GET mdel PICTURE '!'
          READ NOUP
          IF mdel='Y'
                 @ 21,5
                 *     * calculate changes
                 STORE p.smh+chg:smh TO chg:smh
                 STORE p.amh+chg:amh TO chg:amh
                 STORE p.smh*p.per+chg:per TO chg:per
                 DELETE
```

```
    ENDIF
  ENDDO
     @ 21,0
     *    * calculate and record changes in ps.dbf
     STORE "Wait.." to note
     @ 21,5 GET note
     CLEAR GETS
     SELECT SECONDARY
     USE ps INDEX ps
     FIND '&msno'
     STORE s.smh TO old:smh
     REPLACE s.smh WITH s.smh-chg:smh NOUPDATE
       IF s.amh>=chg:amh
          REPLACE s.amh WITH s.amh-chg:amh NOUPDATE
       ENDIF
     STORE s.smh TO new:smh
     STORE s.actper TO old:per
     REPLACE s.actper WITH ((old:per*old:smh)-chg:per)/new:smh NOUPDATE
     RELEASE chg:amh,chg:smh,chg:per,old:smh,old:per,new:smh
     SELECT SECOND
     USE pdf INDEX pdf
STORE 'Y' TO mprn
@ 21,0
@ 21,5 say 'Do you want a printout for this Task? ' ;
GET mprn PICTURE '!'
READ NOUP
IF !(mprn)='Y'
  STORE '4' TO mcmd
  STORE 'B' TO prn
ENDIF
RELEASE note,mprn


CASE mcmd= '4'
                  *       * print out task data
                  *       * and recalculate total schedule
                  *       * actual manhours and percent complete

   STORE 'Y' TO mprint
   IF prn='A'
   @ 6, 0 SAY '                                                    '
   @ 6, 20 SAY 'Do you want a print-out? ' GET mprint PICTURE '!'
   READ NOUP
   ENDIF
   CLEAR GETS

   *      * display IWPR data for a task
   SELECT SECONDARY
   GO TOP
   FIND '&msno'
   STORE bn TO mbn
   STORE title TO mtitle
```

```
         STORE loc TO mloc
         STORE sdate TO msdate
         STORE cdate TO mcdate
         STORE 'A' TO fix
         DO WHILE fix='A'
         *       * get total scheduled manhours
         USE ps INDEX ps
         FIND '&msno'
         STORE s.smh TO old:smh
         USE pdf INDEX pdf
         SELECT PRIMARY
         FIND '&msno'
         *        * printout or display ?
         IF !(mprint)= 'Y'
            STORE 'say' TO gs
            STORE 55 TO maxrow
            SET FORMAT TO PRINT
         ELSE
            STORE 'get' TO gs
            STORE 23 TO maxrow
            SET FORMAT TO SCREEN
            ERASE
         ENDIF !(mprint)= 'Y'
         *       * variables for total schedule and actual manhours
         ST    0 TO tamh,tsmh
               0.00 TO tper
             ILE p.sno=msno.AND.(.NOT.EOF)
            STORE 7 TO row
@ 1, 5 SAY "Schedule No. :"
@ 1,20 &gs msno
@ 1,40 SAY "Title      :"
@ 1,52 &gs mtitle
@ 2, 5 SAY "Battalion    :"
@ 2,20 &gs mbn
@ 2,40 SAY "Location  :"
@ 2,52 &gs mloc
@ 3,5  SAY "Start Date   :"
@ 3,20 &gs msdate
@ 3,40 SAY "Compl Date:"
@ 3,52 &gs mcdate
@ 5, 0 SAY "    I      J      Description    SCH Mhours   ACT Mhours    %WT     % Comp"
@ 6, 0 SAY " ====== ====== ================ ==========  ===========  =======  ======="

             DO WHILE p.sno=msno.AND.(.NOT.EOF).AND.row<maxrow
                @ row,2  SAY p.i USING '9999'
                @ row,9 SAY p.j USING '9999'
                @ row,16 SAY $(p.des,1,16)
                @ row,36 SAY p.smh USING '99,999'
                @ row,49 SAY p.amh USING '99,999'
                @ row,59 SAY (p.smh/old:smh)*100.00 USING '999.99'
                @ row,68 SAY p.per USING '999'
                STORE p.amh+tamh TO tamh
                STORE p.smh+tsmh TO tsmh
                STORE ((p.smh/old:smh)*p.per)+tper TO tper
                STORE row+2 TO row
                SKIP +1
             ENDDO WHILE p.sno=msno.AND.(.NOT.EOF).AND.row<maxrow
             IF (p.sno<>msno).OR.EOF
@ row,0   SAY "-----------------------------------------------------------------------------
@ row+1,16 SAY 'TOTALS:'
@ row+1,36 SAY tsmh USING '999,999'
@ row+1,49 SAY tamh USING '999,999'
@ row+1,68 SAY tper USING '999.99'
              ENDIF
              IF !(mprint)= 'Y'
                 EJECT
              ELSE
                 STORE 'Press any key to continue' TO note
                 @ 23,20 GET note
                 CLEAR GETS
                 SET CONSOLE OFF
                 WAIT
                 SET CONSOLE ON
                 ERASE
               ENDIF !(mprint)= 'Y'

            ENDDO WHILE p.sno=msno.AND.(.NOT.EOF)

            IF !(mprint)='Y'
               EJECT
               SET FORMAT TO SCREEN
            ENDIF
            *   * store correct totals in ps.dbf
            SELECT SECOND
            USE ps INDEX ps
            FIND '&msno'
            REPLACE s.smh WITH tsmh,s.amh WITH tamh,s.actper WITH tper NOUPDATE
            USE pdf INDEX pdf
            *   * tell user error was found
            *   * old schedule manhours incorrect
            *   * must regenerate report
            IF (tsmh<>old:smh)
                ERASE
                @ 10,0 SAY CHR(7)+CHR(7)+CHR(7)+CHR(7)
                ? 'INCONSISTENT DATA FOUND : IT HAS BEEN CORRECTED'
                ? 'DISREGARD LAST REPORT   : CORRECT REPORT FOLLOWS'
            ELSE
                STORE 'B' TO fix
                RELEASE mbn,mtitle,mloc,msdate,mcdate,tamh,tper,tsmh,row
                RELEASE old:smh
            ENDIF
       ENDDO
```

```
      CASE mcmd= '5'

         *        * display tasks that have 1WPR data
         SELECT PRIMARY
         GOTO TOP
         IF #=0
            @ 22,10 SAY '                                  '
            @ 22,10 SAY 'There are NO IWPR's in the data base '
            STORE 1 TO COUNT
            DO WHILE COUNT<100
               STORE COUNT+1 TO COUNT
            ENDDO WHILE COUNT<100
         ELSE
            SELECT SECONDARY
            GOTO TOP
            IF #=0
               @ 21,10 SAY '                                      '
               STORE 'There are NO tasks in Task Data Base!' TO note
               @ 21,10 GET note
               CLEAR GETS
               STORE 1 TO COUNT
               DO WHILE COUNT<100
                  STORE COUNT+1 TO COUNT
               ENDDO WHILE COUNT<100
             ELSE
                STORE s.sno TO msno
                STORE 'Y' TO mprint
                @ 22,10 SAY '                                             '
                @ 22,10 SAY 'Do you want a print-out? (y/n) ' ;
                GET mprint PICTURE '!'
                READ NOUP
                CLEAR GETS
                @ 22,10 SAY '                                         '
                STORE 'WAIT...PROCESSING' TO note
                @ 22,10 GET note
                CLEAR GETS
                SELECT SECONDARY
                IF !(mprint)= 'Y'
                   SET FORMAT TO PRINT
                   STORE 62 TO maxrow
                ELSE
                   SET FORMAT TO SCREEN
                   STORE 24 TO maxrow
                   ERASE
                ENDIF !(mprint)= 'Y'

                DO WHILE .NOT.EOF
@ 1,23 SAY "Tasks that HAVE IWPR data"
@ 3,5 SAY "Sch. No.  Battalion        Title                    Location"
@ 4,5 SAY "========  =========  ======================    ================ "
                   STORE 6 TO row

                   DO WHILE .NOT. EOF.AND.row<maxrow
                SELECT PRIMARY
                FIND '&msno'
                IF #<>0
                   @ row,6 SAY s.sno
                   @ row,18 SAY s.bn
                   @ row,28 SAY s.title
                   @ row,54 SAY s.loc
                   STORE row+2 TO row
                ENDIF #<>0
                SELECT SECONDARY
                SKIP
                STORE s.sno TO msno
             ENDDO WHILE .NOT. EOF.AND.row<maxrow

             IF !(mprint)= 'Y'
                EJECT
             ELSE
                STORE 'Y' TO con
                @ 23,20 SAY "Continue ? " GET con PICTURE '!'
                READ NOUP
                CLEAR GETS
                IF !(con)='Y'
                  ERASE
                ELSE
                  GO BOTT
                  SKIP
                ENDIF
             ENDIF !(mprint)= 'Y'
        ENDDO WHILE .NOT.EOF

        IF !(mprint)='Y'
          EJECT
          SET FORMAT TO SCREEN
        ENDIF

     ENDIF #=0

  ENDIF #=0

CASE mcmd= '6'
   *       * display tasks not having IWPR data
   SELECT SECONDARY
   GOTO TOP
   IF #=0
      @ 22,10 SAY '                                        '
      @ 22,10 SAY 'There are NO tasks in Project Data Base!'
      STORE 1 TO COUNT
      DO WHILE COUNT<100
         STORE COUNT+1 TO COUNT
      ENDDO WHILE COUNT
   ELSE
      SELECT PRIMARY
```

```
            GOTO TOP
            IF #=0
               @ 22,10 SAY '                                     '
               STORE "There are NO IWPR's in the Data Base!" TO note
               @ 22,10 GET note
               CLEAR GETS
               STORE 1 TO COUNT
               DO WHILE COUNT<100
                  STORE COUNT+1 TO COUNT
               ENDDO WHILE COUNT<100
            ELSE
               STORE 'Y' TO mprint
               @ 22,10 SAY '                                            '
               @ 22,10 SAY 'Do you want a print-out? (y/n) ';
               GET mprint PICTURE '!'
               READ NOUP
               CLEAR GETS
               @ 22,10 SAY '                                        '
               STORE 'WAIT...PROCESSING' TO note
               @ 22,10 GET note
               CLEAR GETS
               IF !(mprint)= 'Y'
                  SET FORMAT TO PRINT
                  STORE 62 TO maxrow
               ELSE
                  SET FORMAT TO SCREEN
                  STORE 24 TO maxrow
                  ERASE
               ENDIF !(mprint)= 'Y'

               SELECT SECONDARY
               GO TOP
               STORE s.sno TO msno
               DO WHILE .NOT.EOF
@ 1,18 SAY "Tasks that have NO IWPR data"
@ 3,5 SAY "Sch. No.  Battalion         Title                    Location"
@ 4,5 SAY "========  =========  ======================    ================= "
                  STORE 6 TO row
                  STORE 0 TO COUNT
                  DO WHILE .NOT. EOF.AND.row<maxrow
                     SELECT PRIMARY
                     FIND '&msno'
                     IF #=0
                        @ row,6 SAY s.sno
                        @ row,18 SAY s.bn
                        @ row,28 SAY s.title
                        @ row,54 SAY s.loc
                        STORE row+2 TO row
                        STORE COUNT+1 TO COUNT
                     ENDIF #=0
                     SELECT SECONDARY
                     SKIP
               STORE s.sno TO msno
            ENDDO WHILE .NOT. EOF.AND.row<maxrow

            IF COUNT=0
               @ 6,15 SAY "ALL Tasks in Project Data Base HAVE IWPR's!"
               IF !(mprint)<> 'Y'
                  STORE 1 TO COUNT
                  DO WHILE COUNT<100
                     STORE COUNT+1 TO COUNT
                  ENDDO WHILE COUNT<100
               ENDIF !(mprint)<> 'Y'
             ENDIF COUNT=0

             IF !(mprint)= 'Y'
                EJECT
             ELSE
                STORE 'Y' TO con
                @ 23,20 SAY "Continue ? " GET con PICTURE '!'
                READ NOUP
                CLEAR GETS
                IF !(con)='Y'
                    ERASE
                ELSE
                    GO BOTT
                    SKIP
                ENDIF
             ENDIF !(mprint)= 'Y'

          ENDDO WHILE .NOT.EOF

          IF !(mprint)='Y'
            EJECT
            SET FORMAT TO SCREEN
          ENDIF
      ENDIF #=0
   ENDIF #=0


CASE mcmd= '0'
   *      * backup data base to B drive
   *      * ask user if backup is needed before exiting program
   STORE 'Y' TO mbak
   @ 22,0 SAY "
   @ 22,10 SAY "Do you want to backup the IWPR database to B drive? " ;
   GET mbak PICTURE '!'
   READ NOUP
   IF !(mbak)= 'Y'
      ERASE
      @ 10,10
      ? "Copying to B: ...Be patient"
      ?
      ?
```

```
            ?
            ?
            SELECT PRIMARY
            USE
            USE act
            COPY STRUCTURE TO b:act
            USE b:act
            INDEX ON sno+STR(i,4,0)+STR(j,4,0) TO b:act
            SET INDEX TO b:act
            APPEND FROM a:act
            ERASE
            RETURN
         ENDIF !(mbak)= 'Y'
         ERASE
         QUIT
   ENDCASE

ENDDO mcmd<> '0'
*
*       * END ACTMNU.CMD
*
```

```
******************************************************************************
*
*
*                           A - C - M - S  Ver. 84.0
*
*                                   USA-CERL
*
*       PROGRAM: LISTPDF.CMD
*
*       DESCRIPTION: This is a subroutine of PDFMNU2.CMD and lists task
*                    data for each schedule number in TPDF.DBF (a tempoary
*                    data base).  User may display on screen (SPPDF.CMD)
*                    or send to printer (PPDF.CMD).
*
*
*       CALLS: PPDF.CMD and SPPDF.CMD, called by PDFMNU2.CMD
*
******************************************************************************
*       *  pdf is project data base
SELECT PRIMARY
USE pdf INDEX pdf
*       * tpdf is a temporary data base
*       * containing a set of schedule numbers
SELECT SECONDARY
USE tpdf
GO BOTTOM
*       * get last record number
STORE # TO no
GO TOP
@ 21,0 SAY STR(no,3,0)+ " tasks found in the database."
STORE 0 TO mstop
STORE 'Y' TO mprint
@ 22,0 SAY "Do you want the print-outs? (y/n) " GET mprint PICTURE '!'
READ
CLEAR GETS
@ 21,0
@ 22,0
IF !(mprint)= "Y"
   *    * send to printer
   SET FORMAT TO PRINT
   DO WHILE .NOT.EOF
         STORE # TO mrec
         STORE sno TO msno
         SELECT PRIMARY
         FIND '&msno'
         *******
         DO ppdf
         *******
         SELECT SECONDARY
         USE tpdf
         GOTO mrec
         SKIP +1
   ENDDO WHILE .NOT.EOF
   EJECT
   SET FORMAT TO SCREEN
ELSE
   *    * display on console
   DO WHILE (.NOT.EOF).AND.(mstop=0)
     STORE # TO mrec
     STORE sno TO msno
     SELECT PRIMARY
     FIND '&msno'
     ********
     DO sppdf
     ********
     SELECT SECONDARY
     GOTO mrec
     USE tpdf
     SKIP +1
   ENDDO WHILE .NOT.EOF
ENDIF !(mprint)= 'Y'
SELECT SECONDARY
USE ps INDEX ps
*       * delete temporary file
DELET FILE tpdf
RETURN
*
*       * END LISTPDF.CMD return to PDFMNU2.CMD
*
```

```
*******************************************************************************
*
*
*                           A - C - M - S  Ver. 84.0
*
*                                      USA-CERL
*
*       PROGRAM: MCTRPT.CMD
*
*       DESCRIPTION: Monthly Contractor and Troop Construction Report.
*                    Produces a report comparing estimated contractor and
*                    troop construction cost for a monthly period.  The period
*                    data are obtained by subtracting the current totals from
*                    totals of the last period.  There are three fields in the
*                    PS.DBF data base used to store the last period data:
*                    pper = last period percent complete, pfunds = total funds
*                    since last report, pdirdays = total mandays since last
*                    report.  Note:  The report can only be generated once.
*
*
*       CALLS: none
*
*******************************************************************************
ERASE
SET DELETE ON
SET COLON OFF
SET TALK OFF
@ 1,0 SAY '                 MONTHLY CONTRACT AND TROOP REPORT'
*       * get printer/console control codes
RESTORE FROM install ADDITIVE
*       * store project types to memory variables
STORE " " TO mpjtype0
STORE "Operations and Maintenance, ARMY" TO mpjtype3
STORE "Minor Military Construction, ARMY" TO mpjtype2
STORE "Military Construction, ARMY" TO mpjtype1
STORE "Minor Maintenance and Repair" TO mpjtype4
STORE "Community Assistance Projects" TO mpjtype5
STORE "Disaster Relief Projects" TO mpjtype6
STORE "Self Help Projects" TO mpjtype7
*       * get date of last processing
SELECT SECONDARY
USE misc
STORE date TO mdate
@ 3,0 SAY 'For Period Ending:' get mdate
CLEAR GETS
STORE 'Y' TO mok
@ 5,0 SAY 'Is the Period Ending Date OK? (y/n) ' get mok picture '!'
READ
CLEAR GETS
IF mok<>'Y'
   RETURN
```

```
ENDIF
@ 7,0
? "Set Printer to Top-of-Form"
? "     Press RETURN to print"
SET CONSOLE OFF
WAIT
SET CONSOLE ON
ERASE
@ 1,0 SAY '                   MONTHLY CONTRACT AND TROOP REPORT'
@ 3,0 SAY '                       For Period Ending:'get mdate
CLEAR GETS
@ 7,29 SAY CHR(27)+'G'+'6'+'PROCESSING'+CHR(27)+'G'+'0'
*
*-----------------------------routine to print report------------------------
*
*       * get week(lastmonth) report was generated and replace with this week
STORE lastmonth TO mlastmonth
REPLACE lastmonth WITH mdate
*       * routine to convert mmddyy to military date: dd MMM yy
STORE 'JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC' TO year
STORE $(mdate,1,2) TO mnth
STORE $(mdate,3,2)+' '+$(year,(val(mnth)*3-2),3)+' '+$(mdate,5,2) TO mil
STORE $(mlastmonth,1,2) TO mnth
STORE $(mlastmonth,3,2)+' '+$(year,(val(mnth)*3-2),3)+' '+$(mlastmonth,5,2) TO
milast
*       * use task data to print report in battalion order
*       * indexed by battalion,project type, and customer
SELECT PRIMARY
USE pdf
INDEX ON !(bn)+pjtype+$(customer,1,8) TO pdf.mon
SET INDEX TO pdf.mon
GOTO TOP
DO WHILE .NOT.EOF
  SET FORMAT TO SCREEN
  STORE bn TO mbn
  @ 9,29 SAY "Battalion:"
  @ 9,41 GET mbn
  CLEAR GETS
  SET FORMAT TO PRINT
  @ 0,0 SAY p12
  @ 1,7 SAY "&dw Monthly Contractor And Troop Construction Report "+p12
  @ 2,86 SAY "Unit: &ep"+bn+"&epoff"
  @ 3,86 SAY "Period: &ep"+milast+" - "+"&mil&epoff"
  @ 5,0 SAY "                      Percent             Total        Total;
    Contractor  Troop Cons. "
  @ 6,0 SAY "                      Actually            Estimated    Estimated;
    Cost ($)    Cost ($)    Direct Cons. Cumula- "
  @ 7,0 SAY "Task  No.             Placed              Contractor   Troop;
    Placed      Placed      Placed       tive    "
  @ 8,0 SAY "& Title               (period/cumulative) Cost ($)     Cost ($);
    For Period  For Period  Man Days     Man Days"
  @ 9,0 SAY "=============================================================;
```

```
===================================================="
  *        * tecost= total estimated contractor cost
  STORE 0 TO tecost
  *        * tetcost= total estimated troop construction cost
  STORE 0 TO tetcost
  *        * tecostl= total contractor cost placed for period
  STORE 0 TO tecostl
  *        * tetcostl= total troop construction cost for period
  STORE 0 TO tetcostl
  *        * tmdperiod = total mandays this period
  STORE 0 TO tmdperiod
  *        * tmdtodate = total mandays to date
  STORE 0 TO tmdtodate
  STORE 10 TO row
  DO WHILE !(bn)=!(mbn).AND.(.NOT.EOF)
    STORE pjtype TO mpjtype
    DO WHILE (pjtype=mpjtype).AND.(bn=mbn).AND.(.NOT.EOF)
        IF row>=58
           STORE 5 TO row
        ENDIF
        @ row,40 SAY mpjtype&mpjtype
        @ row+1,0 SAY "======================================================;
============================================================="
        STORE row+2 TO row
        STORE $(customer,1,10) TO mcustomer
        STORE customer TO mcustomer2
        *        * mecost = sub total estimated contractor cost
        STORE 0 TO mecost
        *        * metcost = sub total estimated troop construction cost
        STORE 0 TO metcost
        *        * metcost = sub total contractor cost for period
        STORE 0 TO mecostl
        *        * metcostl = sub total troop construction cost for period
        STORE 0 TO metcostl
        *        * subtmdp = sub total mandays this period
        STORE 0 TO subtmdp
        *        * sub total mandays to date
        STORE 0 TO subtmdc
        DO WHILE !($(customer,1,10))=!(mcustomer).AND.(pjtype=mpjtype).AND.;
        (bn=mbn).AND.(.NOT.EOF)
           IF row>=60
                STORE 6 TO row
           ENDIF
           @ row,0 SAY sno
           STORE sno TO msno
           SELECT SECONDARY
           *    * get mandays
           USE mandays INDEX mandays
           FIND "&msno"
           STORE s.ocharge+s.ncharge+s.echarge TO dirtot
           *    * ps keeps cumulatives:
           *    * pper = percent complete as of last report
           *    * pfunds = total funds since last report
           *    * pdirdays = total mandays since last report
           USE ps INDEX ps
           FIND '&msno'
           STORE pper TO mpper
           STORE pfunds TO mpfunds
           STORE pdirdays TO mpdir
           *    * mpdir = monthly mandays
           STORE dirtot-mpdir TO mpdir
           *    * save for next report
           REPLACE pdirdays WITH dirtot
           *    * current percent complete
           STORE actper TO mactper
           *    * percent complete for period
           STORE mactper-mpper TO percentp
           *    * save for next report
           REPLACE pper WITH mactper
           *    * actual troop cost to date
           STORE ec1+ec2+ec3+ec4+ec5 TO mtcost
           *    * troop cost this period
           STORE mtcost-pfunds TO mpfunds
           *    * save for next report
           REPLACE pfunds WITH mtcost
           *    * print monthly data for task
           @ row+1,0 SAY $(p.title,1,20)
           @ row+1,22 SAY percentp USING "999.9%"
           @ row+1,31 SAY mactper  USING "999.9%"
           @ row+1,43 SAY p.ecost USING "$9999999999"
           @ row+1,55 SAY p.etcost USING "$9999999999"
           @ row+1,67 SAY p.ecost*percentp/100 USING "$9999999999"
           @ row+1,79 SAY mpfunds USING "$9999999999"
           @ row+1,95 SAY mpdir USING "999999999"
           @ row+1,104 SAY dirtot USING "999999999"
           @ ROW+2,0 SAY p.pno
           *    * accumulate sub totals
           STORE p.ecost+mecost TO mecost
           STORE p.etcost+metcost TO metcost
           STORE (p.ecost*percentp/100)+mecost1 TO mecost1
           STORE mpfunds+metcost1 TO metcost1
           STORE dirtot+subtmdc TO subtmdc
           STORE mpdir+subtmdp TO subtmdp
           @ row+3,0 SAY "---------------------------------------------------;
-----------------------------------------------------------------"
           STORE row+4 TO row
           SELECT PRIMARY
           SKIP 1
        ENDDO
        *       * print sub totals
        @ row,0 SAY "&ep"+"Sub Total &mcustomer2  . . ."
        @ row,44 SAY mecost USING "$9999999999"
        @ row,56 SAY metcost USING "$9999999999"
        @ row,68 SAY mecost1 USING "$9999999999"
        @ row,80 SAY metcostl USING "$9999999999"
        @ row,96 SAY subtmdp USING "999999999"
        @ row,105 SAY subtmdc USING "999999999"
        @ row+1,0 SAY
"&epoff=================================================;
================================================================"
        *       * accumulate totals
        STORE mecost+tecost TO tecost
        STORE metcost+tetcost TO tetcost
        STORE mecostl+tecostl TO tecostl
        STORE metcostl+tetcostl TO tetcostl
        STORE tmdperiod+subtmdp TO tmdperiod
        STORE tmdtodate+suutmdc TO tmdtodate
        STORE row+2 TO row
    ENDDO
  ENDDO
  *     * print totals
  @ row,0 SAY "&ep"+"Total &mbn"
  @ row,44 SAY tecost USING "$9999999999"
  @ row,56 SAY tetcost USING "$9999999999"
  @ row,68 SAY tecostl USING "$9999999999"
  @ row,80 SAY tetcostl USING "$9999999999"
  @ row,96 SAY tmdperiod USING "999999999"
  @ row,105 SAY tmdtodate USING "999999999"
  @ row+2,0 SAY
"&epoff=====================================================;
========================================================"
  IF row<30
     @ 50,0 SAY '    '
     EJECT
  ENDIF
ENDDO
@ 1,0 SAY CHR(30)
SET FORMAT TO SCREEN
SELECT PRIMARY
USE
DELETE FILE pdf.mon
ERASE
RETURN
*
*       * END MCTRPT.CMD
*
```

```
*******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                       USA-CERL
*
*        PROGRAM: PDF.CMD
*
*        DESCRIPTION: Initiates menu fuction 1 of PDFMNU.CMD.  This is where
*                     a task enters the system.  Each task is identified
*                     by a unique six-digit schedule number.  This routine
*                     provides for entering and changing task data.  Main
*                     data base is PDF.DBF, but allocated funds are stored in
*                     in PS.DBF.
*
*        CALLS: PDF1.CMD, called by and returns to PDFMNU.CMD
*
*******************************************************************************
*
ERASE
*       * use project data base
*       * set up default control memory variables
STORE "XXXXXX" TO msno
STORE 'Y' TO mok
STORE 'Y' TO mmore

DO WHILE !(mmore)= 'Y'
   SELECT PRIMARY
   ERASE
   *    * get schedule number of task
   @ 13,0 SAY "please enter the SCHEDULE NUMBER " ;
   GET msno PICTURE "XXXXXX"
   READ
   CLEAR GETS

   FIND '&msno'
   IF #=0
      @ 16,0 SAY "Is this a NEW task in the database? (y/n)" ;
      GET mok PICTURE '!'
      READ
      CLEAR GETS
      IF !(mok)= 'Y'
         APPEND BLANK
         REPLACE sno WITH msno
         REPLACE pjtype WITH "0"
         REPLACE graf WITH "Y"
         *******
         DO pdf1
         *******
      ELSE
```

```
         @ 16,0 SAY '                                                  '
         @ 16,0 SAY "Task NOT found, please check."
      ENDIF !(mok)= 'Y'
   ELSE
      *     * show user header information
      ERASE
      SET COLON OFF
      @ 2,20 SAY "PROJECT DATA FORM (page 1)"
      @ 4, 0 SAY "PN   Number: " GET pno
      @ 4,38 SAY "SCHEDULE NUMBER: " GET msno
      @ 6, 0 SAY "Title      : " GET title
      @ 6,38 SAY "Location        : " GET loc
      @ 8, 0 SAY "Battalion  : " GET bn
      @ 8,38 SAY "Project Type    : " GET pjtype
      CLEAR GETS
      @ 11, 0 SAY "Check above CAREFULLY, is this the correct task? (y/n)" ;
      GET mok PICTURE '!'
      READ
      CLEAR GETS
      IF !(mok)= 'Y'
         *******
         DO pdf1
         *******
      ENDIF !(mok)= 'Y'
      CLEAR GETS

   ENDIF #=0
   ERASE
   STORE "Do you want to continue entering project data forms? (y/n)" TO mnote
   @ 10,5 GET mnote
   CLEAR GETS
   @ 10,66 GET mmore PICTURE '!'
   READ
   ERASE
ENDDO WHILE !(mmore)= 'Y'

*       * returns to PDFMNU.CMD if mmore <> Y

RETURN
*
*       * END PDF.CMD
*
```

```
*****************************************************************************
*
*
*                         A - C - M - S  Ver. 84.0
*
*                                USA-CERL
*
*       PROGRAM: PDF1.CMD
*
*       DESCRIPTION: Subroutine called by PDF.CMD for entering task header
*                    information.  Inherits msno (schedule number)
*                    and PDF.DBF record cursor.
*
*
*       CALLS: PDF2.CMD, called by PDF1.CMD returns to PDF.CMD
*
*****************************************************************************
*
ERASE
SET COLON OFF
@ 2,19 SAY "PROJECT DATA FORM (page 1)"
@ 4,0 SAY "SCHEDULE NUMBER: " GET msno
CLEAR GETS
@ 4,38 SAY "PN  Number: " GET pno
@ 6, 0 SAY "Title     : " GET title
@ 6,38 SAY "Location       : " GET loc
@ 8, 0 SAY "BN/CO/PLT :     / / "
@ 8,12 GET bn PICTURE '9999'
@ 8,17 GET co PICTURE 'A'
@ 8,19 GET plt PICTURE '9'
@ 8,38 SAY "Project Type   : " GET pjtype
@ 10, 0 SAY "Est. Contractor Cost  : " GET ecost
*       * display possible task types
@ 10,38 SAY "+-----  Types of Tasks ------------------+"
@ 11,38 SAY "| 0)  NOT SPECIFIED YET                  |"
@ 12, 0 SAY "Est. Troop Const. Cost: "
@ 12,24 GET etcost
@ 12,38 SAY "| 1)  Military Construction, ARMY        |"
@ 13,38 SAY "| 2)  Minor Military Construction, ARMY  |"
@ 14, 0 SAY "Start  Date (mmddyy)  : " GET sdate
@ 14,38 SAY "| 3)  Operations & Maintenance, ARMY     |"
@ 15,38 SAY "| 4)  Minor Maintenance & Repair         |"
@ 16, 0 SAY "Compl. Date (mmddyy)  : " GET cdate
@ 16,38 SAY "| 5)  Community Assistance Projects      |"
@ 17,38 SAY "| 6)  Self Help Projects                 |"
@ 18, 0 SAY "Customer : " GET customer
@ 18,38 SAY "+----------------------------------------+"
@ 21, 0 SAY "-----------------------------------------------------------------
;
--------------"
@ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field;
```

```
      ;  CTR-D: next character ;"
@ 23, 5 SAY "CTR-W: next screen ;  CTR-S: previous character ;"

READ
CLEAR GETS
STORE 'Y' TO graf
@ 20,0 SAY "Is this a Graf project? (y/n)" GET graf PICTURE '!'
READ
CLEAR GETS
IF !(graf)<> "Y"
   *    * set non-Graf TDY rates for task
   @ 20,0
   @ 20,0 SAY "Set TDY Rate      OFF:         NCO:         EM:        "
   @ 20,23 GET off PICTURE '99.99'
   @ 20,36 GET nco PICTURE '99.99'
   @ 20,48 GET em  PICTURE '99.99'
   READ
   CLEAR GETS
ENDIF !(graf)= "N"

*******
DO pdf2
*******

RETURN
*
*       * END PDF1.CMD
*
```

```
******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                 USA-CERL
*
*       PROGRAM: PDF2.CMD
*
*       DESCRIPTION: This routine gets task funding data for Project Data
*                    Base.  All funds are stored in PDF.DBF except for al-
*                    located funds, which are stored in PS.DBF.  Inherits
*                    msno and record cursor in PDF.DBF.
*
*
*
*       CALLS: by PDF1.CMD
*
******************************************************************************
*
ERASE
SET COLON OFF
STORE bn TO mbn
@ 1,19 SAY "PROJECT DATA FORM (page 2)"
@ 2, 0 SAY "SCHEDULE NO : " GET sno
@ 2,22 SAY "Title: " GET title
@ 2,48 SAY "PN  No. : " GET pno
CLEAR GETS
*        * esc# is for estimated funds
@  3, 0 SAY "+---------------+---------------+---------------+"
@  4, 0 SAY "|  TYPES OF     |   ESTIMATED   |   ALLOCATED   |"
@  5, 0 SAY "|  FUNDS        |   COST        |   FUNDS       |"
@  6, 0 SAY "+---------------+---------------+---------------+"
@  7, 0 SAY "|  US TDY       |"
@  7,17 GET esc1
@  7,31 SAY "|               |"
@  8, 0 SAY "+---------------+---------------+---------------+"
@  9, 0 SAY "|  US EQUIP     |"
@  9,17 GET esc2
@  9,31 SAY "|               |"
@ 10, 0 SAY "+---------------+---------------+---------------+"
@ 11, 0 SAY "|  CSG TDY      |"
@ 11,17 GET esc3
@ 11,31 SAY "|               |"
@ 12, 0 SAY "+---------------+---------------+---------------+"
@ 13, 0 SAY "|  CSG EQUIP    |"
@ 13,17 GET esc4
@ 13,31 SAY "|               |"
@ 14, 0 SAY "+---------------+---------------+---------------+"
@ 15, 0 SAY "|  CSG LABOR    |"
@ 15,17 GET esc5
@ 15,31 SAY "|               |"
@ 16, 0 SAY "+---------------+---------------+---------------+"
@ 17, 0 SAY "|  TOTAL        |"
@ 17,31 SAY "|               |"
@ 18, 0 SAY "+---------------+---------------+---------------+"
@ 21, 0 SAY "----------------------------------------------------------------;
---------------"
@ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field;
     ;  CTR-D: next character ;"
@ 23, 5 SAY "CTR-W: next screen ;  CTR-S: previous character ;"

READ
CLEAR GETS
STORE escl+esc2+esc3+esc4+esc5 TO mesc
@ 17,17 GET mesc
CLEAR GETS

*       * NOTE allocated funds are stored in ps.dbf
SELECT SECOND
USE ps INDEX ps
FIND '&msno'
*       * set up associated data bases for new project
IF #=0
   REPLACE sno WITH msno
   USE mandays INDEX mandays
   APPEND BLANK
   REPLACE sno WITH msno
   USE ps INDEX ps
   APPEND BLANK
   REPLACE sno WITH msno
ENDIF #=0
*       * make any changes to organization of project
REPLACE s.bn WITH p.bn,s.co WITH p.co,s.plt WITH p.plt
*       * ac# is for allocated funds
@  7,33 GET acl
@  9,33 GET ac2
@ 11,33 GET ac3
@ 13,33 GET ac4
@ 15,33 GET ac5
READ
STORE acl TO macl
STORE ac2 TO mac2
STORE ac3 TO mac3
STORE ac4 TO mac4
STORE ac5 TO mac5
SELECT SECOND
FIND '&msno'
REPLACE acl WITH macl,ac2 WITH mac2,ac3 WITH mac3,ac4 WITH mac4,ac5 WITH mac5
STORE acl+ac2+ac3+ac4+ac5 TO mac
@ 17,33 GET mac
CLEAR GETS
@ 20, 0 SAY "COMMENTS:"
@ 20,10 GET comments
READ
CLEAR GETS
*       * set up battalion cost data base
*       * one record per battalion
USE bn INDEX bn
find '&mbn'
IF #=0
APPEND BLANK
REPLACE bn WITH mbn,billno WITH 1,from WITH 'XXXXXX'
ENDIF
SELECT SECOND
USE ps INDEX ps
*       * return to PDF1.CMD
RETURN
*
*       *  END PDF2.CMD
*
```

```
******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                 USA-CERL
*
*       PROGRAM: PDFMNU.CMD
*
*       DESCRIPTION: Implements the Project Data Base.  Tasks must enter the
*                    system through the Project Data Base.  Main data base
*                    used is PDF.DBF.  Allows user to enter and change task
*                    data display what tasks are in the data base, delete
*                    task, and set default TDY rates.  Function 4 calls
*                    submenu for selective report generation.
*
*
*       CALLS: PDF.CMD and PDFMNU2.CMD
*
******************************************************************************
*
SET TALK OFF
SET COLON OFF
SET DELETE ON
STORE '1' TO mcmd

DO WHILE mcmd<> '6'
   *    * set up memory variables
   STORE 'XXXXXX' TO msno
   STORE '0000' TO bn
   STORE 'XXXXXXXXXXXXXX' TO mdirno
   STORE '1' TO mcmd
   SELECT PRIMARY
   USE pdf INDEX pdf
   SELECT SECONDARY
   USE ps INDEX ps
   ERASE
   @  2,10 SAY "***********************************************************"
   @  3,10 SAY "*               PROJECT DATA BASE MENU                    *"
   @  4,10 SAY "***********************************************************"
   @  5,10 SAY "*                                                         *"
   @  6,10 SAY "*  <1> Enter/Change Task Data                             *"
   @  7,10 SAY "*                                                         *"
   @  8,10 SAY "*  <2> Delete A Tasks                                     *"
   @  9,10 SAY "*                                                         *"
   @ 10,10 SAY "*  <3> Display/Print Schedule No.'s For All Tasks         *"
   @ 11,10 SAY "*                                                         *"
   @ 12,10 SAY "*  <4> Display/Print Data For Selected Tasks              *"
   @ 13,10 SAY "*                                                         *"
   @ 14,10 SAY "*  <5> Set Graf TDY Rates OFF:_____ NCO:_____ EM:_____    *"
   @ 15,10 SAY "*          (Current Rates =                            )  *"
   @ 16,10 SAY "*                                                         *"
   @ 17,10 SAY "*  <6> Exit                                               *"
   @ 18,10 SAY "*                                                         *"
   @ 19,10 SAY "***********************************************************"

   *    * retrieve Graf TDY rates
   SELECT SECOND
   USE misc
   GOTO 1
   @ 15,40 GET goff
   @ 15,50 GET gnco
   @ 15,59 GET gem
   CLEAR GETS

   @ 21,10 SAY "OPTION? "
   @ 21,17 GET mcmd PICTURE '9'
   READ
   CLEAR GETS

   SELECT PRIMARY
   *    * primary is always pdf

   DO CASE

      CASE mcmd= '1'
         *      * enter/change task data
         ******
         DO pdf
         ******

      CASE mcmd= '2'
         *      * delete a task
         @ 21,9 SAY " Enter Schedule No. to Delete " ;
         GET msno PICTURE "XXXXXX"
         READ
         CLEAR GETS
         STORE 'N' TO msure
         @ 21,47 SAY "Sure? (Y/N) " GET msure PICTURE '!'
         READ
         CLEAR GETS
         IF !(msure) = 'N'
            @ 21,0 SAY "                                               "
         ELSE
            IF !(msure)= 'Y'
               FIND '&msno'
               IF #=0
                  @ 21,0 SAY
"                                                 "

                  @ 21,20 GET msno
                  CLEAR GETS
                  @ 21,28 SAY "is NOT in Task Data Base"
               ELSE
```

```
                 *     * delete the sno from all the involved databases
                 STORE "          WAIT...DELETING                           "
;
                 TO mnote
                 @ 21,10 GET mnote
                 @ 21,37 GET msno
                 CLEAR GETS
                 DELETE
                 PACK
                 SELECT SECOND
                 USE ps INDEX ps
                 FIND '&msno'
                 DELETE
                 PACK
                 USE schedule INDEX schedule
                 DELETE FOR sno=msno
                 PACK
                 USE groups INDEX groups
                 DELETE FOR sno=msno
                 PACK
                 @ 21,0 SAY "                                         "
              ENDIF #=0
           ENDIF !(msure)= 'N'
        ENDIF !(msure)= 'Y'


     CASE mcmd = '3'
        *      * display on console or print one line
        *      * discription of all tasks in data base
        SELECT PRIMARY
        GOTO TOP
        IF #=0
           STORE "There are NO projects in the Data Base" TO mnote
           @ 21,20 GET mnote
           CLEAR GETS
           STORE 1 TO mark
           DO WHILE mark<100
              STORE mark+1 TO mark
           ENDDO WHILE mark<100
        ELSE
           STORE 'Y' TO mnew
           @ 21,5 SAY "                      "
           @ 21,20 SAY " Do you want print-outs? " GET mnew PICTURE '!'
           READ
           CLEAR GETS

           IF !(mnew)= 'Y'
              SET FORMAT TO PRINT
              STORE 60 TO row
              DO WHILE (.NOT. EOF )
@ 2,10 SAY "Schedule Numbers Currently In Project DataBase"
@ 4,0 SAY "Sch. No.  Battalion          Title
```

```
Location        "
 @ 5,0 SAY "========  =========  ======================
============================="
                  STORE 7 TO row2
                  DO WHILE (row2<=row).AND.(.NOT.EOF)
                     @ row2,1 SAY sno
                     @ row2,13 SAY bn
                     @ row2,23 SAY title
                     @ row2,49 SAY loc
                     STORE row2+2 TO row2
                     SKIP
                  ENDDO WHILE (row2<=row).AND.(.NOT.EOF)
                  EJECT
               ENDDO WHILE (.NOT. EOF )
               EJECT
               SET FORMAT TO SCREEN
               @ 21,20 SAY "
"
            ELSE
               IF !(mnew)= 'N'
                  SET FORMAT TO SCREEN
                  STORE 21 TO row
                  STORE 0 TO mstop
                  STORE 'Y' TO mok
                  DO WHILE (.NOT. EOF ).AND.(mstop=0)
                     ERASE
@ 1,15 SAY "Schedule Numbers Currently In Project DataBase"
@ 3,5 SAY "Sch. No.  Battalion          Title
Location        "
@ 4,5 SAY "========  =========  ======================
==========================="
                     STORE 6 TO row2
                     DO WHILE (row2<=row).AND.(.NOT.EOF)
                        @ row2,6 SAY sno
                        @ row2,18 SAY bn
                        @ row2,28 SAY title
                        @ row2,54 SAY loc
                        STORE row2+2 TO row2
                        SKIP
                     ENDDO WHILE (row2<=row).AND.(.NOT.EOF)

                     @ 23,15 SAY  "Do you wish to continue? " GET mok
                     READ
                     CLEAR GETS
                     IF mok<>'Y'
                     STORE 1 TO mstop
                     LOOP
                     ENDIF
                     @ 23,0
                  ENDDO WHILE (.NOT. EOF )
                  @ 21,0
               ENDIF !(mnew)= 'N'
            ENDIF !(mnew)= 'Y'
         ENDIF #=0

      CASE mcmd= '4'
         *      * submenu for selective display/print of projects
         **********
         DO pdfmnu2
         **********

      CASE mcmd= '5'
         *      * set default TDY rates
         @ 21,0 SAY "                                      "
         SELECT SECONDARY
         USE misc
         GOTO 1
         @ 14,40 GET goff PICTURE '99.99'
         @ 14,50 GET gnco PICTURE '99.99'
         @ 14,59 GET gem  PICTURE '99.99'
         READ
         CLEAR GETS

      CASE mcmd= '6'
         *      * exit program with option to backup to B drive
         STORE 'Y' TO mbak
         @ 21,0 SAY
"                                                                      "
         @ 21,10 SAY "Do you want to backup the project database to B drive?
(y/n)" GET mbak PICTURE '!'
         READ
         IF !(mbak)= 'Y'
            ERASE
            @ 10,0
            ? "WAIT.. COPYING PROJECT DATA TO B DRIVE"
            ?
            ?
            ?
            ?
            SELECT PRIMARY
            USE
            SELECT SECONDARY
            USE
            USE pdf
            COPY STRUCTURE TO b:pdf
            USE b:pdf
            INDEX ON sno TO b:pdf
            SET INDEX TO b:pdf
            APPEND FROM a:pdf
            USE ps
            COPY STRUCTURE TO b:ps
            USE b:ps
            INDEX ON sno TO b:ps
            SET INDEX TO b:ps
            APPEND FROM a:ps
            USE
            ERASE
            RETURN
         ENDIF !(mbak)= 'Y'
         ERASE
         RETURN

   ENDCASE

ENDDO WHILE mcmd<> '6'
*
*       * END PDFMNU.CMD
*
```

```
*******************************************************************************
*
*
*                           A - C - M - S  Ver. 84.0
*
*                                   USA-CERL
*
*       PROGRAM: PDFMNU2.CMD
*
*       DESCRIPTION: Submenu implementing function 4 on main menu of Project
*                    Data Base.  Provides for selective task data reports
*                    based on task number, funding type, battalion, or
*                    for a single task.  The structure is a simple DO CASE.
*
*
*       CALLS: LISTPDF.CMD, PPDF.CMD, SPPDF.CMD, returns to PDFMNU.CMD
*
*******************************************************************************
SET DELETE ON
DO WHILE mcmd<> '5'
   STORE '1' TO mcmd
   ERASE
   @  2,10 SAY "***************************************************"
   @  3,10 SAY "*       DISPLAY/PRINT DATA FOR SELECTED TASK      *"
   @  4,10 SAY "***************************************************"
   @  5,10 SAY "*                                                 *"
   @  6,10 SAY "*                                                 *"
   @  7,10 SAY "*  <1> All Tasks of a PN Number                   *"
   @  8,10 SAY "*                                                 *"
   @  9,10 SAY "*  <2> All Tasks of a Funding Type                *"
   @ 10,10 SAY "*                                                 *"
   @ 11,10 SAY "*  <3> All Tasks of a Battalion                   *"
   @ 12,10 SAY "*                                                 *"
   @ 13,10 SAY "*  <4> For a Schedule Number                      *"
   @ 14,10 SAY "*                                                 *"
   @ 15,10 SAY "*  <5> Return to Main Menu                        *"
   @ 16,10 SAY "*                                                 *"
   @ 17,10 SAY "*                                                 *"
   @ 18,10 SAY "***************************************************"

   *    * set up for display of task types
   STORE "MCA" TO mpjtype1
   STORE "MMCA" TO mpjtype2
   STORE "OMA" TO mpjtype3
   STORE "MM&R" TO mpjtype4
   STORE "ComAsst" TO mpjtype5
   STORE "DisRelief" TO mpjtype6
   STORE "SelfHelp" TO mpjtype7
   STORE "NOT SPECIFIED" TO mpjtype0
   *    * get user's choice
   @ 20,10 SAY "OPTION? "
@ 20,17 GET mcmd PICTURE '9'
READ
CLEAR GETS
*    * all task data is in pdf.dbf (except allocated funds in ps.dbf)
SELECT PRIMARY
USE pdf INDEX pdf

DO CASE

   CASE mcmd= '1'
      *      * display or print task data for all
      *      * tasks funded on a directive number
      STORE 'XXXXXXXXXXXXXX' TO mdirno
      @ 20,5 SAY "                 "
      @ 20,10 SAY "Enter PN Number: " GET mdirno PICTURE "XXXXXXXXXXXXXX"
      READ
      CLEAR GETS
      *      * create temp data base of schedule numbers for report
      COPY TO tpdf FIELD sno FOR pno=mdirno
      **********
      DO listpdf
      **********

   CASE mcmd= '2'
      *      * display or print task data for all
      *      * task of a funding type
      *      * display the funding types on the screen
      @ 7,64 SAY "Funding Types"
      @ 8,64 SAY "1) MCA"
      @ 9,64 SAY "2) MMCA"
      @ 10,64 SAY "3) OMA"
      @ 11,64 SAY "4) MM&R"
      @ 12,64 SAY "5) ComAsst"
      @ 13,64 SAY "6) DisRelief"
      @ 14,64 SAY "7) SelfHelp"
      @ 15,64 SAY "0) UNSPECIFIED"

      STORE '1' TO mpjtype
      @ 20,5 SAY "                   "
      @ 20,10 SAY "Enter Funding Type " GET mpjtype PICTURE '9'
      READ
      CLEAR GETS
      DO WHILE mpjtype> '7' .OR.mpjtype< '0'
         @ 20,10 SAY "Enter Funding Type " GET mpjtype PICTURE '9'
         READ
         CLEAR GETS
      ENDDO WHILE mpjtype> '7' .OR.mpjtype< '0'
      *      * create temp data base
      COPY FIELD sno TO tpdf FOR pjtype=mpjtype
      **********
      DO listpdf
      **********
```

```
CASE mcmd= '3'
   *      * display or print task data
   *      * for all tasks of a battalion
   STORE '0000' TO mbn
   @ 20,5 SAY "                      "
   @ 20,10 SAY "Enter Battalion " GET mbn PICTURE '9999'
   READ
   CLEAR GETS
   * create temp data base
   COPY FIELD sno TO tpdf FOR bn=mbn
   **********
   DO listpdf
   **********

CASE mcmd= '4'
   *      * display or print task data
   *      * for a task ( selected by schedule number)
   STORE 'XXXXXX' TO msno
   USE pdf INDEX pdf
   @ 20,5 SAY "                   "
   @ 20,10 SAY "Enter Schedule No. " GET msno PICTURE "XXXXXX"
   READ
   CLEAR GETS

   FIND '&msno'
   IF #=0
      @ 20,10 GET msno
      CLEAR GETS
      @ 20,16 SAY "  is NOT in Project Data Base."
      STORE "Press any key to continue" TO mnote
      @ 22,10 GET mnote
      CLEAR GETS
      SET CONSOLE OFF
      WAIT
      SET CONSOLE ON
   ELSE
      STORE 'Y' TO mnew
      @ 21,15 SAY "Do you want a print-out? " GET mnew PICTURE '!'
          READ
      CLEAR GETS
      IF !(mnew)= 'Y'
         SET FORMAT TO PRINT
         *******
         DO ppdf
         *******
         EJECT
         SET FORMAT TO SCREEN
      ELSE
         IF !(mnew)= 'N'
            ********
            DO sppdf
```

```
                  ********
               ENDIF !(mnew)= 'N'
            ENDIF !(mnew)= 'Y'
         ENDIF #=0

      CASE mcmd= '5'
         *      * return to PDFMNU.CMD
         RETURN

  ENDCASE

ENDDO WHILE !(mcmd)<> '5'
*
*       * END PDFMNU2.CMD
*
```

```
*****************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: PPDF.CMD
*
*       DESCRIPTION: Subroutine to print Project Data Form on line printer.
*                    Called by PDFMNU2.CMD to print one Project Data Form,
*                    or by LISTPDF.CMD for functions 1 through 3 Project Data
*                    Base report generation submenu.
*
*


*
*       CALLS: called by PDFMNU2.CMD or LISTPDF.CMD
*
*****************************************************************************
STORE 'JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC' TO year
SET MARGIN TO 6
@ 1,15 SAY "*****  T A S K **** D A T A  *****"
@ 2, 0 SAY "PN   Number:"
@ 2,13 SAY pno
@ 2,38 SAY "Schedule Number:"
@ 2,55 SAY sno
STORE sno TO msno
@ 3, 0 SAY "Title:"
@ 3, 7 SAY title
@ 3,38 SAY "Loc.:"
@ 3,44 SAY loc
@ 4, 0 SAY "Battalion:"
@ 4,11 SAY bn
@ 4,38 SAY "Project Type:"
STORE pjtype TO pj
@ 4,53 SAY mpjtype&pj
@ 5, 0 SAY "Esti. Contractor Cost  :"
@ 5,26 SAY ecost
@ 6, 0 SAY "Esti. Troop Const. Cost:"
@ 6,26 SAY etcost
store sdate to mdate
*       * routine to convert mmddyy to military date: dd MMM yy
store $(mdate,1,2) to mnth
store $(mdate,3,2)+' '+$(year,(val(mnth)*3-2),3)+' '+$(mdate,5,2) to ms
@ 7, 0 SAY "Start Date     : "+"&ms"
store cdate to mdate
store $(mdate,1,2) to mnth
store $(mdate,3,2)+' '+$(year,(val(mnth)*3-2),3)+' '+$(mdate,5,2) to mc
@ 8, 0 SAY "Compl. Date    : "+"&mc"
@ 9, 0 SAY "Customer:"
@ 9,10 SAY customer
SELECT SECONDARY
USE ps INDEX ps
FIND '&msno'
*       * allocated funds
STORE ac1 TO mac1
STORE ac2 TO mac2
STORE ac3 TO mac3
STORE ac4 TO mac4
STORE ac5 TO mac5
*       * expended funds
STORE ec1 TO mec1
STORE ec2 TO mec2
STORE ec3 TO mec3
STORE ec4 TO mec4
STORE ec5 TO mec5
SELECT PRIMARY
FIND '&msno'
@ 11, 0 SAY "+---------------+---------------+---------------+--------------+"
@ 12, 0 SAY "|  TYPES OF     |   ESTIMATED   |   ALLOCATED   |   ACTUAL     |"
@ 13, 0 SAY "|  FUNDS        |   COST        |   FUNDS       |   EXPENDED   |"
@ 14, 0 SAY "+---------------+---------------+---------------+--------------+"
@ 15, 0 SAY "|  US TDY       |"
@ 15,17 SAY esc1
@ 15,31 SAY "|"
@ 15,33 SAY mac1
@ 15,47 SAY "|"
@ 15,50 say mec1
@ 15,62 say "|"
@ 16, 0 SAY "+---------------+---------------+---------------+--------------+"
@ 17, 0 SAY "|  US EQUIP     |"
@ 17,17 SAY esc2
@ 17,31 SAY "|"
@ 17,33 SAY mac2
@ 17,47 SAY "|"
@ 17,50 SAY mec2
@ 17,62 say "|"
@ 18, 0 SAY "+---------------+---------------+---------------+--------------+"
@ 19, 0 SAY "|  CSG TDY      |"
@ 19,17 SAY esc3
@ 19,31 SAY "|"
@ 19,33 SAY mac3
@ 19,47 SAY "|"
@ 19,50 say mec3
@ 19,62 say "|"
@ 20, 0 SAY "+---------------+---------------+---------------+--------------+"
@ 21, 0 SAY "|  CSG EQUIP    |"
@ 21,17 SAY esc4
@ 21,31 SAY "|"
@ 21,33 SAY mac4
@ 21,47 SAY "|"
@ 21,50 say mec4
@ 21,62 say "|"
@ 22, 0 SAY "+--------------+---------------+---------------+--------------+"
@ 23, 0 SAY "|  CSG LABOR   |"
@ 23,17 SAY esc5
@ 23,31 SAY "|"
@ 23,33 SAY mac5
@ 23,47 SAY "|"
@ 23,50 say mec5
@ 23,62 say "|"
@ 24, 0 SAY "+--------------+---------------+---------------+--------------+"
@ 25, 0 SAY "|  TOTAL       |"
STORE mac1+mac2+mac3+mac4+mac5 TO mac
STORE esc1+esc2+esc3+esc4+esc5 TO mesc
STORE mec1+mec2+mec3+mec4+mec5 TO mec
@ 25,17 SAY mesc
@ 25,31 SAY "|"
@ 25,33 SAY mac
@ 25,47 SAY "|"
@ 25,50 say mec
@ 25,62 say "|"
@ 26, 0 SAY "+--------------+---------------+---------------+--------------+"
@ 30, 0 SAY "COMMENTS:"
@ 30,10 SAY $(comments,1,40)
@ 32,10 SAY $(comments,41,29)
EJECT
*       * return to PDFMNU2.CMD or LISTPDF.CMD
RETURN
*
*       * END PPDF.CMD
*
```

```
******************************************************************************
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: PS.CMD
*
*       DESCRIPTION: This program processes the Weekly Progress Reports.
*                    The transaction data bases for the multiuse WPR
*                    input are merged into sinlge data bases and cleared
*                    for next week's input.  For each WPR entered, the master
*                    record(s) of that project is updated.
*
*       CALLS: none
*
******************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
SET DELETE ON
*RESTORE FROM install ADDITIVE
*-------------------------------WARNING-----------------------------
* to users of a mulit-user system were file locks may have been
* set on W*A.DBF or W*B.DBF type files due to multi-user WPR.CMD input
* a system reboot will clear all locks: otherwise BDOS ERROR FILE R/O
@ 7,0
?? '                         ***NOTE***'
?
?  '       You Must Reboot--turn off and turn back on the system'
?
?  "       Before Processing the WPR's"
?
STORE 'N' TO mdone
@ 15,8 SAY '          Have You Done So ?' GET mdone PICTURE '!'
READ
IF !(mdone)<>'Y'
   QUIT
ENDIF
ERASE
RELEASE mdone
*-------------------------------MERGE TRANSACTION DATA BASES----------
*
@ 1,0 SAY '                  PROCESS WEEKLY PROGRESS REPORTS'
@ 12,0
?? '                           Merging Data Bases'
* merge user-a and user-b w*.dbf files
SELECT PRIMARY
USE wpr INDEX wpr
APPEND FROM wpra
APPEND FROM wprb
```

```
USE wmandays INDEX wmandays
APPEND FROM wmdaysa
APPEND FROM wmdaysb
USE wact INDEX wact
APPEND FROM wacta
APPEND FROM wactb
USE a:weq INDEX weq
APPEND FROM weqa
APPEND FROM weqb
* reset (clear) working databases for next week's input
* create temporary data bases: tw*a.dbf and tw*b.dbf
*
USE wpra
COPY STRUCTURE TO twpra
COPY STRUCTURE TO twprb
USE twpra
INDEX ON sno+week TO wpra
USE twprb
INDEX ON sno+week TO wprb
*
USE wmdaysa
COPY STRUCTURE TO twmdaysa
COPY STRUCTURE TO twmdaysb
USE twmdaysa
INDEX ON sno+week TO wmdaysa
USE twmdaysb
INDEX ON sno+week TO wmdaysb
*
USE wacta
COPY STRUCTURE TO twacta
COPY STRUCTURE TO twactb
USE twacta
INDEX ON sno+week+STR(i,4,0)+STR(j,4,0) TO wacta
USE twactb
INDEX ON sno+week+STR(i,4,0)+STR(j,4,0) TO wactb
*
USE weqa
COPY STRUCTURE TO tweqa
COPY STRUCTURE TO tweqb
USE tweqa
INDEX ON sno+week TO weqa
USE tweqb
INDEX ON sno+week TO weqb
USE
* delete old w*a.dbf files
DELETE FILE wpra.dbf
DELETE FILE wmdaysa.dbf
DELETE FILE wacta.dbf
DELETE FILE weqa.dbf
* rename empty temp files for next week
RENAME twpra TO wpra
RENAME twmdaysa TO wmdaysa
RENAME twacta TO wacta
RENAME tweqa TO weqa
*delete old w*b.dbf files
DELETE FILE wprb.dbf
DELETE FILE wmdaysb.dbf
DELETE FILE wactb.dbf
DELETE FILE weqb.dbf
* rename temp files
RENAME twprb TO wprb
RENAME twmdaysb TO wmdaysb
RENAME twactb TO wactb
RENAME tweqb TO weqb
* * * * * * * * * * * * * * * * MAIN PROCESSING ROUTINE * * * * * * * * * * * * *
ERASE
USE misc
* get graff tdy rates
STORE goff TO goff
STORE gnco TO gnco
STORE gem TO gem
* get date of last processing
STORE date TO mdate
@ 1,0 SAY '                     PROCESS WEEKLY PROGRESS REPORTS'
USE days INDEX days
* days contains fridays
FIND '&mdate'
* get this week
SKIP +1
STORE day TO mdate
STORE mdate TO mdate1
STORE "N" TO msure
* give user option to process for any period
DO WHILE !(msure)<> "Y"
   @ 4,0 SAY "Enter date of the period end(Wednesday): " GET mdate
   @ 5,0 SAY '     or press RETURN if correct'
   READ
   CLEAR GETS
   IF mdate<>mdate1
      FIND '&mdate'
      IF #=0
         @ 6,0 SAY 'Period Ending date must be Wednesday.  Enter again.'
         STORE mdate1 TO mdate
      ELSE
         @ 5,0
         @ 6,0
         @ 6,0 SAY "Sure? (y/n) " GET msure PICTURE '!'
         READ
         CLEAR GETS
      ENDIF #=0
   ELSE
      STORE "Y" TO msure
   ENDIF mdate<>mdate1
ENDDO WHILE !(msure)<> "Y"
```

PS.CMD

```
USE misc
* store processing date for next time
REPLACE date WITH mdate
SELECT PRIMARY
USE ps INDEX ps
SELECT SECOND
USE wpr INDEX wpr
@ 12,0
@ 12,30 SAY rv  +'PROCESSING'+ rvoff

DO WHILE .not.EOF
   STORE s.sno TO msno
   STORE # TO rec
   SELECT PRIMARY
   FIND "&msno"
   IF p.date>=mdate
      SELECT SECOND
      FIND "&msno"
      DO WHILE (s.sno=msno).AND.(.not.EOF)
                 SKIP +1
      ENDDO WHILE (s.sno=msno).AND.(.not.EOF)
      LOOP
   ELSE
      SELECT SECOND
      FIND "&msno"
      * skip previously processsed wprs
      DO WHILE (s.sno=msno).AND.(week<=p.date).AND.(.not.EOF)
         SKIP +1
      ENDDO WHILE (s.sno=msno).AND.(week<=p.date).AN
      * this loop processes all wpr ' s up to the desired date.
      DO WHILE (s.sno=msno).AND.(s.week>p.date).AND.(s.week<=mdate)
         @ 14,20 SAY "        Schedule No.:" GET msno
         CLEAR GETS
         STORE sno+week TO msnoweek
         SELECT PRIMARY
         REPLACE p.date WITH s.week NOUPDATE
         REPLACE p.cws WITH s.cws NOUPDATE
         REPLACE p.adjust WITH s.adjust NOUPDATE
         REPLACE ec5 WITH ec5+s.csglbr NOUPDATE
         REPLACE ec4 WITH ec4+s.csgeq NOUPDATE
         REPLACE ec3 WITH ec3+s.csgtdy NOUPDATE
         REPLACE ec1 WITH ec1+s.other NOUPDATE
         REPLACE ec2 WITH ec2+s.rental+s.trans NOUPDATE
         *--------------------------------
         *      * processing mandays data
         *--------------------------------
         SELECT SECOND
         USE pdf INDEX pdf
         FIND "&msno"
         IF !(graf)= "N"
            *           * for non-graf projects, find the particular tdy rates
            STORE OFF TO ngoff
            STORE nco TO ngnco
            STORE em TO ngem
            USE wmandays INDEX wmandays
            FIND "&msnoweek"
            SELECT PRIMARY
            * primary is ps
            REPLACE ecl WITH
ecl+s.ocharge*ngoff+s.ncharge*ngnco+s.echarge*ngem
         ELSE
            USE wmandays INDEX wmandays
            FIND "&msnoweek"
            SELECT PRIMARY
            REPLACE ecl WITH ecl+s.ocharge*goff+s.ncharge*gnco+s.echarge*gem
NOUPDATE
         ENDIF !(graf)= "N"
         *----------------------------------
         *       * processing equipment data
         *----------------------------------
         SELECT PRIMARY
         USE eqcost INDEX eqcost
         SELECT SECOND
         USE weq INDEX weq
         FIND "&msnoweek"
         *       * use useqcost to store us equip cost.
         STORE 0 TO useqcost
         STORE 0 TO csceqcost
         * loop through all the weq records of this wpr
         DO WHILE (s.sno+s.week=msnoweek).AND.(.NOT.EOF)
            STORE "1" TO no
            * looping through all 4 equip items in one weq record
            DO WHILE no<= "4"
             IF s.eno&no<>0
               IF s.type&no= "O"
                  STORE str(&no+1,1,0) TO no
                  LOOP
               ENDIF
               STORE str(s.eno&no,4,0)+str(s.atchno&no,2,0) TO mkey
               SELECT PRIMARY
               FIND "&mkey"
               IF s.type&no='U'
                 STORE p.omcost*s.hr&no +useqcost TO useqcost
               ELSE
                 STORE p.omcost*s.hr&no+csceqcost TO csceqcost
               ENDIF
             ENDIF s.eno&no<>0
             STORE str(&no+1,1,0) TO no
            ENDDO WHILE no<= "4"
            SELECT SECOND
            SKIP +1
         ENDDO WHILE (sno+week=msnoweek).AND.(.not.EOF)
         *--------------------------------
```

```
*      * processing activity data
*------------------------------
STORE 0 TO new:per, new:amh, new:smh
SELECT SECOND
USE wact INDEX wact
FIND '&msnoweek'
IF #<>0
SELECT PRIMARY
USE act INDEX act
FIND '&msno'
DO WHILE (p.sno=msno).AND.(.NOT.EOF)
   STORE msnoweek+STR(p.i,4,0)+STR(p.j,4,0) TO mij
   SELECT SECOND
   FIND '&mij'
   IF (#<>0).AND.(s.amh<>0).AND.(s.per<>0)
   SELECT PRIMARY
   REPLACE p.amh WITH p.amh+s.amh NOUPDATE
   REPLACE p.per WITH s.per NOUPDATE
   ENDIF
   SELECT PRIMARY
   STORE p.smh+new:smh TO new:smh
   STORE p.amh+new:amh TO new:amh
   STORE p.smh*p.per+new:per TO new:per
   SKIP +1
ENDDO
STORE (new:per/new:smh) TO new:per
ENDIF
*----------------------------
*      * update mandays master
*----------------------------
SELECT SECOND
USE wmandays INDEX wmandays
FIND "&msnoweek"
SELECT PRIMARY
USE mandays INDEX mandays
FIND "&msno"
REPLACE p.osupport WITH p.osupport+s.osupport NOUP
REPLACE p.nsupport WITH p.nsupport+s.nsupport NOUP
REPLACE p.esupport WITH p.esupport+s.esupport NOUP
REPLACE p.ocharge  WITH p.ocharge+s.ocharge   NOUP
REPLACE p.owork    WITH p.owork+s.owork       NOUP
REPLACE p.ncharge  WITH p.ncharge+s.ncharge   NOUP
REPLACE p.nwork    WITH p.nwork+s.nwork       NOUP
REPLACE p.echarge  WITH p.echarge+s.echarge   NOUP
REPLACE p.ework    WITH p.ework+s.ework       NOUP
REPLACE p.olost WITH p.olost+s.olost NOUP
REPLACE p.mlost WITH p.mlost+s.mlost NOUP
REPLACE p.wlost WITH p.wlost+s.wlost NOUP
*-----------------------------------------------
*      * accumulate useq cost,csceq cost and per comp
*-----------------------------------------------
SELECT PRIMARY
```

```
         USE ps INDEX ps
         FIND "&msno"
         REPLACE p.ec2 WITH ec2+useqcost NOUP
         REPLACE p.ec4 WITH ec4+csceqcost NOUP
         IF new:per > 0
         REPLACE p.actper WITH new:per NOUP
         ENDIF
         IF new:amh > 0
         REPLACE p.amh WITH new:amh NOUP
         ENDIF
         IF new:smh > 0
         REPLACE p.smh WITH new:smh NOUP
         ENDIF
         RELEASE new:per,new:amh,new:smh
         * get next wpr to process
         SELECT SECOND
         USE wpr INDEX wpr
         FIND "&msno"
         DO WHILE (sno=msno).AND.(week<=p.date).AND.(.not.EOF)
            SKIP 1
         ENDDO WHILE (sno=msno).AND.(week<=p.date).AND.
      ENDDO WHILE (s.sno=msno).AND.(s.week>p.date).a
   ENDIF p.date>=mdate
*       * make sure program does not get hung in loop
SELECT SECONDARY
IF #=rec
  SKIP +1
ENDIF

ENDDO WHILE .not.EOF

ERASE
QUIT
*
*       * END PS.CMD
*
```

```
********************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                    USA-CERL
*
*       PROGRAM: PSR.CMD
*
*       DESCRIPTION: This routine provides for selective generation of
*                    Project Status Reports.  Reports may be generated by
*                    Project Clusters, Groups, or for a single task.
*                    The program gets the user's selection and creates
*                    a temporary data base of schedule numbers for the
*                    reports to be generated, then loops through calling
*                    PSR.CMD to print.
*
*       CALLS: PSR.CMD to print report
*
********************************************************************************
SET TALK OFF
SET COLON OFF
SET DELETE ON
ERASE
*       * get printer/console control codes
RESTORE FROM install ADDITIVE
*       * set up memory variables
STORE 0 TO mcno
STORE 0 TO mgno
STORE 0 TO mxno
STORE "Y" TO ALL
STORE 'Y' TO ftpeo
STORE 'JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC' to year
USE misc
*       * get date of last processing
STORE date TO mdate
@ 1,0 SAY "                  PROJECT STATUS REPORT"
@ 3,0 SAY "Period Ending: " GET mdate PICTURE "999999"
CLEAR GETS
@ 4,0 SAY "Print Reports for JUST those projects processed on ABOVE date?
(y/n) " ;
GET ftpeo PICTURE "!"
*       * 'ftpeo' means 'For This Period End Only'
READ
CLEAR GETS
IF !(ftpeo)<>'Y'
   @ 4,0 SAY "All projects selected below (incl. those NOT active this week)
will be printed."
ENDIF
@ 6,0 SAY "Do you want to print out status reports for ALL the following
```

```
clusters: (y/n) " ;
GET all PICTURE "!"
USE clusters INDEX clusters
GOTO TOP
*       * display clusters for user selection
DO WHILE .NOT.EOF
   @ cno+7,2 SAY cno
   @ cno+7,3 SAY '>'
   @ cno+7,5 SAY cname
   SKIP
ENDDO WHILE .NOT.EOF
READ
CLEAR GETS

IF !(ALL)<> 'Y'
   @ cno+9,7 SAY "which cluster? " GET mcno PICTURE "9"
   @ cno+10,7 SAY "   0=EXIT"
   READ
   CLEAR GETS
   IF mcno=0
      RETURN
   ENDIF mcno=0
   STORE str(mcno,1,0) TO mkey
   FIND "&mkey"
   STORE cname TO mcna
   USE groups INDEX groups
   FIND "&mkey"
   IF #=0
      ? "                 NO Groups in Cluster"
      ? "                   Returning to Menu"
      *         * delay to give user time to read message
      STORE 1 TO count
      DO WHILE count<150
         STORE count+1 TO count
      ENDDO WHILE count<150
      RETURN
   ENDIF #=0

   *    * next phase of report generation
   *    * display groups for user selection
   ERASE
   @ 1,0 SAY "             PROJECT STATUS REPORT"
   @ 3,0 SAY "Do you want reports for ALL the following groups "
   @ 4,0 SAY "within this cluster? (y/n)" GET ALL PICTURE '!'
   @ 6,17 SAY mcna
   DO WHILE (.NOT.EOF).AND.cno=mcno
      @ gno+6,1 SAY gno
      @ gno+6,3 SAY ">"
      @ gno+6,5 SAY gname
      SKIP
   ENDDO WHILE (.NOT.EOF).AND.cno=mcno
   READ
   CLEAR GETS

   IF !(ALL)<> "Y"
      SKIP -1
      @ gno+8,7 SAY "which group? " GET mgno PICTURE '99'
      @ gno+9,7 SAY "   0=EXIT   "
      READ
      CLEAR GETS
      IF mgno=0
         RETURN
      ENDIF mgno=0

      STORE str(mcno,1,0)+str(mgno,2,0) TO mkey2
      FIND '&mkey2'
      STORE gname TO mgna
      USE schedules INDEX schedules

      FIND '&mkey2'
      IF #=0
         ? "                 NO Schedule Numbers "
         ? "                  Returning to Menu"
         STORE 1 TO count
         DO WHILE count<150
            STORE count+1 TO count
         ENDDO WHILE count<150
         RETURN
      ENDIF #=0

      *         * final phase: display schedule numbers for user selection
      ERASE
      @ 1,0 SAY "                 PROJECT STATUS REPORT"
      @ 3,0 SAY "Do you want Project Status Reports for ALL the following
schedules "
      @ 4,0 SAY "within this group? (y/n)" GET ALL PICTURE '!'
      @ 6,24 SAY mgna
      DO WHILE (.NOT.EOF).AND.cno=mcno.AND.gno=mgno
         IF xno<15
            @ xno+6,1 SAY str(xno,2,0)+ '> ' +sno
         ELSE
            IF xno<29.AND.xno>=15
               @ xno-8,16 SAY str(xno,2,0)+ '> ' +sno
            ELSE
               IF xno<42.AND.xno>=29
                  @ xno-22,31 SAY str(xno,2,0)+ '> ' +sno
               ELSE
                  IF xno<56.AND.xno>=42
                     @ xno-34,46 SAY str(xno,2,0)+ '> ' +sno
                  ENDIF xno<56.AND.xno>=42
               ENDIF xno<42.AND.xno>=29
            ENDIF xno<29.AND.xno>=15
         ENDIF xno<15
         SKIP 1
      ENDDO WHILE (.NOT.EOF).AND.cno=mcno.AND.gno=mgno
      READ
      CLEAR GETS

      IF !(ALL)<> "Y"
         @ 22,27 SAY "which schedule? " GET mxno PICTURE '99'
         @ 23,27 SAY "   0=EXIT   "
         READ
         CLEAR GETS
         IF mxno=0
            RETURN
         ENDIF mxno=0

         STORE str(mcno,1,0)+str(mgno,2,0)+str(mxno,2,0) TO mkey3
         FIND '&mkey3'
         IF #=0
            ? "                 NO  Such Schedule Number "
            ? "                  Returning to Menu"
            STORE 1 TO count
            DO WHILE count<150
               STORE count+1 TO count
            ENDDO WHILE count<150
            RETURN
         ENDIF #=0
         *      *  create temporary data base - tsno.dbf of schedule numbers
         COPY TO tsno FOR (cno=mcno).AND.(gno=mgno).AND.(xno=mxno) FIELD sno
      ELSE
      COPY TO tsno FOR (cno=mcno).AND.(gno=mgno) FIELD sno
      ENDIF !(ALL)<> "Y"
   ELSE
      USE schedules INDEX schedules
      COPY TO tsno FOR cno=mcno FIELD sno
   ENDIF !(ALL)<> "Y"
ELSE
   USE schedules INDEX schedules
   COPY TO tsno FIELD sno
ENDIF !(ALL)<> 'Y'

@ 23,30 SAY rv+ 'PROCESSING' +rvoff

SELECT PRIMARY
USE ps INDEX ps
SELECT second
USE tsno
GOTO TOP
*       * go through all schedule numbers in tsno.dbf
*       * calling PSR.CMD to print Project Status Report
DO WHILE .NOT.EOF
   STORE sno TO msno
   SELECT PRIMARY
   USE ps INDEX ps
   FIND "&msno"
```

```
  IF (!(ftpeo) <> 'Y' ).OR.(date>=mdate)

    ********
    DO psrpt
    ********

  ENDIF (ftpeo <> 'Y' ).OR.(date>=mdate)
  SELECT SECONDARY
  USE tsno
  LOCATE NEXT 5000 FOR sno=msno
  SKIP +1
  DO WHILE s.sno=msno.AND.(.NOT.EOF)
   SKIP +1
  ENDDO
ENDDO WHILE .NOT.EOF
ERASE
RETURN
*
*
* *         END PSRPT.CMD
*
```

```
*******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: PSRPT.CMD
*
*       DESCRIPTION: Prints Project Status Report.  This report is at
*                    the task (schedule number) level and a copy should
*                    go to the project officer.  The report prints scheduled
*                    and actual percent complete, cumulative and period
*                    mandays worked, and allocated and expended funds, and
*                    lists all equipment reported as used on the task
*                    for that period.
*
*       CALLS: called by PSRPT.CMD
*
*******************************************************************************
SET FORMAT TO PRINT
SET MARGIN TO   0
*       * get printer/console control codes
RESTORE FROM install ADDITIVE
STORE p.date TO mweek
STORE adjust TO madjust
STORE cws TO mcws
STORE actper TO mactper
store $(mweek,1,2) to mnth
store $(mweek,3,2)+' '+$(year,(val(mnth)*3-2),3)+' '+$(mweek,5,2) to mweek2
SELECT PRIMARY
USE pdf INDEX pdf
FIND '&msno'
STORE loc TO mloc
STORE sdate TO msdate
store $(msdate,1,2) to mnth
store $(msdate,3,2)+' '+$(year,(val(mnth)*3-2),3)+' '+$(msdate,5,2) to msdate
STORE cdate TO mcdate
store $(mcdate,1,2) to mnth
store $(mcdate,3,2)+' '+$(year,(val(mnth)*3-2),3)+' '+$(mcdate,5,2) to mcdate
STORE title TO mtitle
STORE bn+"/"+co+"/"+STR(plt,1,0) to munit
SELECT PRIMARY
USE a:mandays INDEX a:mandays
FIND '&msno'
*       * get mandays for this reporting period
SELECT SECONDARY
USE a:wmandays INDEX a:wmandays
FIND '&msno&mweek'
SET MARGIN TO 10
@  1,0 SAY p10
@  1,9 SAY dw+"PROJECT STATUS REPORT"+p17
@  3, 0 SAY "Section I: Project Identification                        For
Period:"
@  3,69 SAY mweek2
@  5, 0 SAY "(1) Task  Number  :"
@  5,20 SAY msno
@  5,57 SAY "(2) Bn/Co/Plt:"
@  5,72 SAY munit
@  6, 0 SAY "(3) Title:"
@  6,11 SAY mtitle
@  6,57 SAY "(4) Location :"
@  6,72 SAY mloc
@  7, 0 SAY "(5) Start Date:"
@  7,16 SAY msdate
@  7,57 SAY "(5) Completion Date:"
@  7,78 SAY mcdate
@  8, 0 SAY "(7) Number of days ahead(+) or behind(-):"
@  8,42 SAY madjust USING '999'
@  9, 0 SAY "(8) Percent Scheduled complete from CWS :"
@  9,42 SAY mcws USING '999.9%'
@ 10, 4 SAY "Percent Actual complete             :"
@ 10,42 SAY mactper USING '999.9%'
@ 13, 0 SAY "Section II: Personnel Use Period/To Date"
@ 15, 0 SAY "+----------------------------------------------------------------
--------------------------+"
@ 16, 0 SAY "|      |     SUPPORT     | |                          DIRECT
CONSTRUCTION                  |"
@ 17, 0 SAY "|      |-----------------| |-------------------------------------
-------------------------|"
@ 18, 0 SAY "| US   |       (9)       | |      (10)      |      (11)      |
|          (12)          |"
@ 19, 0 SAY "| TDY  | TDY MANDAYS NOT | |  TDY MANDAYS   |    MANDAYS     |
|        MANDAYS         |"
@ 20, 0 SAY "| RATE | CHARGED TO JOB  | |    CHARGED     |    WORKED      |
|        LOST            |"
@ 21, 0 SAY "|      |                 | |    TO JOB      |    ON SITE     |
|        DUE TO          |"
@ 22, 0 SAY "|------|-----------------| |-----------------------------------|
|------------------------|"
@ 23, 0 SAY "|      | Period | To Date| | Period | To Date| Period |To Date |
|       | Peroid |To Date |"
@ 24, 0 SAY "|------+--------+--------| |--------+--------+--------+--------|
|-------+--------+--------|"
@ 25, 0 SAY "| OFF  |        |        | |        |        |        |        |
| Weath |        |        |"
@ 26, 0 SAY "|      |"
@ 26, 8 SAY s.osupport
@ 26,16 SAY "|"
@ 26,17 SAY p.osupport
@ 26,25 SAY "| |"
@ 26,28 SAY s.ocharge
@ 26,36 SAY "|"
@ 26,37 SAY p.ocharge
@ 26,45 SAY "|"
@ 26,46 SAY s.owork
@ 26,54 SAY "|"
@ 26,55 SAY p.owork
@ 26,63 SAY "| |       |"
@ 26,74 SAY s.wlost
@ 26,82 SAY "|"
@ 26,83 SAY p.wlost
@ 26,91 SAY "|"
@ 27, 0 SAY "|------+--------+--------| |--------+--------+--------+--------|
|-------+--------+--------|"
@ 28, 0 SAY "| NCO  |        |        | |        |        |        |        |
| Maint |        |        |"
@ 29, 0 SAY "|      |"
@ 29, 8 SAY s.nsupport
@ 29,16 SAY "|"
@ 29,17 SAY p.nsupport
@ 29,25 SAY "| |"
@ 29,28 SAY s.ncharge
@ 29,36 SAY "|"
@ 29,37 SAY p.ncharge
@ 29,45 SAY "|"
@ 29,46 SAY s.nwork
@ 29,54 SAY "|"
@ 29,55 SAY p.nwork
@ 29,63 SAY "| |       |"
@ 29,74 SAY s.mlost
@ 29,82 SAY "|"
@ 29,83 SAY p.mlost
@ 29,91 SAY "|"
@ 30, 0 SAY "|------+--------+--------| |--------+--------+--------+--------|
|-------+--------+--------|"
@ 31, 0 SAY "| EM   |        |        | |        |        |        |        |
| Other |        |        |"
@ 32, 0 SAY "|      |"
@ 32, 8 SAY s.esupport
@ 32,16 SAY "|"
@ 32,17 SAY p.esupport
@ 32,25 SAY "| |"
@ 32,28 SAY s.echarge
@ 32,36 SAY "|"
@ 32,37 SAY p.echarge
@ 32,45 SAY "|"
@ 32,46 SAY s.ework
@ 32,54 SAY "|"
@ 32,55 SAY p.ework
@ 32,63 SAY "| |       |"
@ 32,74 SAY s.owork
@ 32,82 SAY "|"
@ 32,83 SAY p.owork
@ 32,91 SAY "|"
@ 33, 0 SAY "+------+--------+--------+-+--------+--------+--------+--------+-
+-------+--------+--------+"
*       * get expended funds
SELECT PRIMARY
USE ps INDEX ps
FIND '&msno'
@ 37, 0 SAY "Financial Progress To Date"
@ 39, 0 SAY "+-----------------+-----------------+-----------------+----------
-------+------------------+"
@ 40, 0 SAY "|                 |                 |
|                 |                  |"
@ 41, 0 SAY "|                 |    ALLOCATED    |   EXPENDED      |
REMAINING     |   FINANCIAL      |"
@ 42, 0 SAY "|   TYPE FUNDS    |    FUNDS ($)    |   TO DATE ($)   |   FUNDS
($)     |   % COMPLETE     |"
@ 43, 0 SAY "|-----------------+-----------------+-----------------+----------
-------+------------------|"
@ 44, 0 SAY "|                 |                 |
|                 |                  |"
@ 45, 0 SAY "|   US TDY        |"
@ 45,20 SAY ac1
@ 45,36 SAY "|"
@ 45,38 SAY ec1
@ 45,54 SAY "|"
@ 45,56 SAY ac1-ec1
@ 45,72 SAY "|"
IF ac1<>0
  @ 45,77 SAY 100*ec1/ac1 USING "999.99%"
ELSE
  @ 45,78 SAY "N/A"
ENDIF
@ 45,91 SAY "|"
@ 46, 0 SAY "|-----------------+-----------------+-----------------+----------
-------+------------------|"
@ 47, 0 SAY "|                 |                 |
|                 |                  |"
@ 48, 0 SAY "|   US EQUIPMENT  |"
@ 48,20 SAY ac2
@ 48,36 SAY "|"
@ 48,38 SAY ec2
@ 48,54 SAY "|"
@ 48,56 SAY ac2-ec2
@ 48,72 SAY "|"
IF ac2<>0
  @ 48,77 SAY 100*ec2/ac2 USING "999.99%"
ELSE
  @ 48,78 SAY "N/A"
ENDIF
@ 48,91 SAY "|"
@ 49, 0 SAY "|-----------------+-----------------+-----------------+----------
-------+------------------|"
@ 50, 0 SAY "|                 |                 |
|                 |                  |"
@ 51, 0 SAY "|   CSC TDY       |"
@ 51,20 SAY ac3
@ 51,36 SAY "|"
@ 51,38 SAY ec3
@ 51,54 SAY "|"
@ 51,56 SAY ac3-ec3
@ 51,72 SAY "|"
IF ac3<>0
  @ 51,77 SAY 100*ec3/ac3 USING "999.99%"
ELSE
  @ 51,78 SAY "N/A"
ENDIF
@ 51,91 SAY "|"
@ 52, 0 SAY "|-----------------+-----------------+-----------------+----------
-------+------------------|"
@ 53, 0 SAY "|                 |                 |
|                 |                  |"
@ 54, 0 SAY "|   CSC EQUIPMENT |"
@ 54,20 SAY ac4
@ 54,36 SAY "|"
@ 54,38 SAY ec4
@ 54,54 SAY "|"
@ 54,56 SAY ac4-ec4
@ 54,72 SAY "|"
IF ac4<>0
  @ 54,77 SAY 100*ec4/ac4 USING "999.99%"
ELSE
  @ 54,78 SAY "N/A"
ENDIF
@ 54,91 SAY "|"
@ 55, 0 SAY "|-----------------+-----------------+-----------------+----------
-------+-------------------"
@ 56, 0 SAY "|                 |                 |
|                 |                  |"
@ 57, 0 SAY "|   CSC LABOR     |"
@ 57,20 SAY ac5
@ 57,36 SAY "|"
@ 57,38 SAY ec5
@ 57,54 SAY "|"
@ 57,56 SAY ac5-ec5
@ 57,72 SAY "|"
IF ac5<>0
  @ 57,77 SAY 100*ec5/ac5 USING "999.99%"
ELSE
  @ 57,78 SAY "N/A"
ENDIF
@ 57,91 SAY "|"
@ 58, 0 SAY "|-----------------+-----------------+-----------------+----------
-------+------------------|"
@ 59, 0 SAY "|                 |                 |
|                 |                  |"
```

```
@ 60, 0 SAY "|   TOTALS        |"
@ 60,20 SAY acl+ac2+ac3+ac4+ac5 USINC "9999999.99"
@ 60,36 SAY "|"
@ 60,38 SAY ecl+ec2+ec3+ec4+ec5 USING "9999999.99"
@ 60,54 SAY "|"
@ 60,56 SAY acl+ac2+ac3+ac4+ac5-ecl-ec2-ec3-ec4-ec5 USING "9999999.99"
@ 60,72 SAY "|"
IF acl+ac2+ac3+ac4+ac5<>0
   @ 60,77 SAY 100*(ecl+ec2+ec3+ec4+ec5)/(acl+ac2+ac3+ac4+ac5) USING "999.99%"
ELSE
   @ 60,78 SAY "N/A"
ENDIF
@ 60,91 SAY "|"
@ 61, 0 SAY "+-----------------+-----------------+-----------------+----------
-;
------+------------------+"
*        * set printer to 12 char/inch
@ 62, 0 SAY p12
EJECT
*        * print equipment used on the task this period
SELECT PRIMARY
USE eqcost INDEX eqcost
*        * use this week's WPR data
SELECT SECOND
USE a:weq INDEX a:weq
STORE msno+mweek to msnoweek
FIND "&msnoweek"
   IF #=0
      SET FORMAT TO SCREEN
      RETURN
   ENDIF
   STORE 12 TO row
   SET MARGIN TO 5
@ 1,12 SAY "Weekly Equipment Cost Report For Period: &mweek2"
@ 3, 6 SAY "Task  No. :"
@ 3,17 SAY msno
@ 3,43 SAY "Title     :"
@ 3,54 SAY mtitle
@ 4, 6 SAY "Bn/Co/Plt :"
@ 4,17 SAY munit
@ 4,43 SAY "Location  :"
@ 4,54 SAY mloc
@ 6,0 SAY '**************************************************************;
***************'
@ 8,0 SAY '      DESCRIPTION               ITEM NO.    $/HOUR      HOURS;
         COST  '
@ 10,0 SAY '--------------------------------------------------------------;
---------------'

   STORE 0 TO mitem
   STORE 0 TO mtotal
   DO WHILE (sno+week=msnoweek).AND.(.NOT.EOF).AND.#<>0
```

```
      STORE "1" TO no
      DO WHILE no<= "4"
         IF s.type&no= "0"
            STORE str(&no+1,1,0) TO no
            LOOP
         ENDIF s.type&no= "0"

         IF s.eno&no=0
            STORE str(&no+1,1,0) TO no
            LOOP
         ENDIF s.eno&no=0

         STORE str(s.eno&no,4,0)+str(s.atchno&no,2,0) TO mkey
         SELECT PRIMARY
         FIND "&mkey"
         STORE p.omcost*s.hr&no TO mitem
         @ row,2 SAY p.despl
         @ row,35 SAY p.eno USING '999'
         @ row,38 SAY "."
         @ row,39 SAY p.atchno USING '99'
         @ row,45 SAY p.omcost USING '$999.99'
         @ row,57 SAY s.hr&no USING '9999'
         @ row,65 SAY mitem USING '$9999999.99'
         STORE row+2 TO row
         STORE mitem+mtotal TO mtotal
         STORE str(&no+1,1,0) TO no
      ENDDO WHILE no<= "4"
      SELECT SECOND
      SKIP
   ENDDO WHILE (sno+week=msnoweek).AND.(.NOT.EOF)
@ row,0 SAY '-------------------------------------------------------------------
;
------------'
@ row+2,50 SAY mtotal USING 'WEEKLY TOTAL   $99999999.99'
   IF row+2<65
      EJECT
   ENDIF row+2<65

ENDDO WHILE .NOT.EOF
SET FORMAT TO SCREEN

* return to PSR.CMD
RETURN
*
*       * END PSRPT.CMD
*
```

```
********************************************************************************
*
*
*                           A - C - M - S  Ver. 84.0
*
*                                       USA-CERL
*
*       PROGRAM: S3RPT.CMD
*
*       DESCRIPTION: This routine generates the "S-3 Weekly Construction Status
*                    Report." This report summarizes construction progress
*                    by group and gives schedule number, title, battalion,
*                    start date, completion date, percent scheduled, percent
*                    actually complete, percent funds expended, and days off
*                    critical path.  Also computes a weighted average over
*                    the group for percent scheduled and percent complete.
*
*       CALLS: None
*
********************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
SET DELETE ON
*       * get printer/console control codes
RESTORE FROM install ADDITIVE
*       * set up default memory variables
STORE 0 TO mcno
STORE 0 TO mgno
STORE "Y" TO ALL
*       * get last processing date
USE misc
STORE date TO mdate
@ 1,0 SAY "                 S-3 WEEKLY CONSTRUCTION STATUS REPORT"
@ 3,6 SAY "For period ending: " GET mdate PICTURE "999999"
CLEAR GETS
@ 5,0 SAY "Do you want to print out ALL the group reports for ALL of the "
@ 6,0 SAY "following clusters: (y/n) " GET ALL PICTURE '!'
*       * display clusters for user selection
USE clusters INDEX clusters
GOTO TOP
DO WHILE .NOT.EOF
   @ cno+7,2 SAY cno
   @ cno+7,3 SAY '>'
   @ cno+7,5 SAY cname
   SKIP
ENDDO WHILE .NOT.EOF
READ
CLEAR GETS

IF !(ALL)<> 'Y'
```

```
   @ cno+9,7 SAY "which cluster? " GET mcno PICTURE "9"
   @ cno+10,7 SAY "   0=EXIT"
   READ
   CLEAR GETS
   IF mcno=0
      RETURN
   ENDIF mcno=0
   STORE str(mcno,1,0) TO mkey
   FIND "&mkey"
   STORE cname TO mcna
   USE groups INDEX groups
   FIND "&mkey"
   IF #=0
      ? "                 NO Groups in Cluster"
      ? "                   Returning to Menu"
      STORE 1 TO count
      DO WHILE count<150
         STORE count+1 TO count
      ENDDO WHILE count<150
      RETURN
   ENDIF #=0
   *       * create temporary file for the cluster
   USE schedule INDEX schedule
   COPY TO tsno FIELD sno,cno,gno FOR cno=mcno
ELSE
   USE schedule INDEX schedule
   COPY TO tsno FIELD sno,cno,gno
ENDIF
USE tsno
*      * get temporary file in cluster and group order
INDEX ON STR(cno,1,0)+STR(gno,2,0) TO tsno
SELECT PRIMARY
USE
USE tsno INDEX tsno
@ 17,0
? ' Set Printer to Top of Form'
? '  Press RETURN when ready'
WAIT
@ 16,0 SAY '                              '
@ 17,0 SAY '                              '
@ 18,0 SAY '                              '
@ 19,0 SAY '                              '
@ 20,0 SAY '                              '
* control codes for tvi 950
@ 23,30 SAY chr(27)+ 'G' + '6' + 'PROCESSING' +chr(27)+ 'G' + '0'

*       * routine to convert mmddyy to military date: dd MMM yy
STORE 'JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC' TO year
STORE $(mdate,1,2) TO mnth
STORE $(mdate,3,2)+ ' ' +$(year,(val(mnth)*3-2),3)+ ' ' +$(mdate,5,2) TO mil
*       * routine to print report
SET FORMAT TO PRINT
```

S3RPT.CMD

```
SET MARGIN TO 0
*      * set print to 12 pitch
@ 0,0 SAY p12
DO WHILE .NOT.EOF
   STORE 0.00 TO tsmh
   STORE 0.00 TO tprog
   STORE 0.00 TO tamh
   STORE str(cno,1,0)+str(gno,2,0) TO mkey
   STORE cno TO mcno
   STORE gno TO mgno
   DO WHILE (p.gno=mgno).AND.(p.cno=mcno).AND.(.NOT.EOF)
      STORE p.sno TO msno
      SELECT SECOND
      USE ps INDEX ps
      FIND '&msno'
      STORE s.smh+tsmh TO tsmh
      SELECT PRIMARY
      SKIP +1
   ENDDO
   SELECT SECOND
   USE groups INDEX groups
   FIND "&mkey"
   STORE gname TO mlgname
@  1,25 SAY "S-3 WEEKLY CONSTRUCTION STATUS REPORT"
@  3, 0 SAY ep+mlgname+epoff
@  3,76 SAY "AS OF"
@  3,82 SAY ep+mil+epoff
@  5, 1 SAY "%WT      PROGRESS TASK                      SCH MH       ACT MH
%SCH     %ACT"
@  6,0 SAY "+--------+--------+-------------------------+------------+-------------+-
-------+--------+"

   STORE 7 TO row
   SELECT PRIMARY
   FIND '&mkey'
   DO WHILE gno=mgno .AND. cno=mcno .AND. (.NOT.EOF)

      IF row>=63
         EJECT
@  1,25 SAY "S-3 WEEKLY CONSTRUCTION STATUS REPORT"
@  3, 0 SAY ep+mlgname+epoff
@  3,76 SAY "AS OF"
@  3,82 SAY ep+mil+epoff
@  5, 1 SAY "%WT      PROGRESS TASK                      SCH MH       ACT MH
%SCH     %ACT"
@  6,0 SAY "+--------+--------+-------------------------+------------+-------------+-
-------+--------+"

         STORE 7 TO row
      ENDIF row>=63

      STORE sno TO msno
      SELECT SECOND
      USE pdf INDEX pdf
      FIND "&msno"
      STORE msno+" "+$(s.title,1,16) TO mtask
      USE ps INDEX ps
      FIND "&msno"
      IF tsmh<>0
        STORE (s.actper*(s.smh/tsmh)) TO mprog
        STORE mprog+tprog TO tprog
        STORE s.amh+tamh  TO tamh
      ELSE
        STORE 0 TO mprog
      ENDIF
@  row, 0 SAY "|"
      IF tsmh<>0
@  row, 3 SAY (s.smh/tsmh)*100.0 USING '999.9'
      ELSE
@  row, 3 SAY '  0.0'
      ENDIF
@  row, 9 SAY "|"
@  row,12 SAY mprog USING '999.9'
@  row,18 SAY "|"
@  row,20 SAY mtask
@  row,44 SAY "|"
@  row,49 SAY s.smh USING '99,999'
@  row,57 SAY "|"
@  row,62 SAY s.amh USING '99,999'
@  row,70 SAY "|"
@  row,73 SAY s.cws USING '999.9'
@  row,79 SAY "|"
@  row,82 SAY s.actper USING '999.9'
@  row,88 SAY "|"
@  row+1,0 SAY "+--------+--------+-------------------------+------------+----------
--+--------+--------+"

      SELECT PRIMARY
      SKIP +1
      STORE row+2 TO row
   ENDDO WHILE gno=mgno .AND. cno=mcno .AND. (.NOT.EOF)
@ row, 0 SAY "|  100.0 |"
@ row,12 SAY tprog USING '999.9'
@ row,18 SAY "|-------------------------|"
@ row,48 SAY tsmh USING '999,999'
@ row,57 SAY "|"
@ row,61 SAY tamh USING '999,999'
@ row,70 SAY "|--------|--------|"
@  row+1, 0 SAY "+--------+--------+-------------------------+------------+---------
---+--------+--------+"

   EJECT
ENDDO WHILE .NOT.EOF
*       * set printer back to 10 pitch
```

```
@ 0,0 SAY p10
SET FORMAT TO SCREEN
USE
*       * delete temporary files
DELETE FILE tsno.dbf
DELETE FILE tsno.ndx
ERASE
RETURN
*
*       * END S3RPT.CMD
*
```

```
********************************************************************************
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: SCHEDULE.CMD
*
*       DESCRIPTION: Implements last level of Project Reporting Groups
*                    Definition program.  Provides for enter, edit, and delete
*                    of schedule numbers in groups.  There may be up to 99
*                    schedule numbers in one group.  Checking is done
*                    to insure valid entries.
*
*       CALLS: called by and returns to GROUPS.CMD
*
********************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
SET DELETE ON
SELECT SECOND
USE pdf INDEX pdf
SELECT PRIMARY
USE schedules INDEX schedules
STORE ' ' TO page
STORE "L" TO cmd

DO WHILE !(cmd)<> 'Q'

   DO CASE

      *        * present page 1 for up to 54 schedule numbers

      CASE !(cmd)= 'L' .AND.page<> 'L'

         STORE 'L' TO page
@  1,21 SAY "SCHEDULE NUMBERS FOR GROUP"
@  2, 2 SAY "   1>                       2>
3>                  "
@  3, 2 SAY "   4>                       5>
6>                  "
@  4, 2 SAY "   7>                       8>
9>                  "
@  5, 2 SAY "  10>                      11>
12>                 "
@  6, 2 SAY "  13>                      14>
15>                 "
@  7, 2 SAY "  16>                      17>
18>                 "
@  8, 2 SAY "  19>                      20>
```

```
21>                     "
@  9, 2 SAY "  22>                       23>
24>                     "
@ 10, 2 SAY "  25>                       26>
27>                     "
@ 11, 2 SAY "  28>                       29>
30>                     "
@ 12, 2 SAY "  31>                       32>
33>                     "
@ 13, 2 SAY "  34>                       35>
36>                     "
@ 14, 2 SAY "  37>                       38>
39>                     "
@ 15, 2 SAY "  40>                       41>
42>                     "
@ 16, 2 SAY "  43>                       44>
45>                     "
@ 17, 2 SAY "  46>                       47>
48>                     "
@ 18, 2 SAY "  49>                       50>
51>                     "
@ 19, 2 SAY "  52>                       53>
54>                     "
@ 20, 2 SAY "--------------------------------------------------------------------
-----"
@ 21, 2 SAY " E to edit/add a schedule no.    N to goto next page    Q to quit this
menu "
@ 22, 2 SAY " D to delete a schedule no.      P to print schedule no.'s          "
@ 23, 2 SAY " OPTION ?"
@ 1,48 GET mgname
         CLEAR GETS

         SELECT PRIMARY
         FIND "&mkey"
         DO WHILE cno=mcno.AND.gno=mgno.AND.(.NOT.EOF).AND.(xno<=54).AND.#<>0
            STORE int(xno/3) TO row
            STORE xno-row*3 TO col
            IF col=0
               STORE 3 TO col
               STORE row-1 TO row
            ENDIF col=0
            @ row+2,25*(col-1)+8 SAY sno
            STORE sno TO msno
            SELECT SECOND
            FIND '&msno'
            IF #=0
               @ row+2,25*(col-1)+15 SAY "UnknownSchNo"
            ELSE
               @ row+2,25*(col-1)+15 SAY $(title,1,13)
            ENDIF #=0
            SELECT PRIMARY
            SKIP
         ENDDO WHILE cno=mcno.AND.gno=mgno.AND.(.NOT.eo
         STORE 'E' TO cmd

      *         * page 2 for 55-99 schedule numbers

      CASE !(cmd)= 'N' .AND.page<> 'N'

         STORE 'N' TO page
@  1,21 SAY "SCHEDULE NUMBERS FOR GROUP"
@  2, 2 SAY "  55>                        56>
57>                     "
@  3, 2 SAY "  58>                        59>
60>                     "
@  4, 2 SAY "  61>                        62>
63>                     "
@  5, 2 SAY "  64>                        65>
66>                     "
@  6, 2 SAY "  67>                        68>
69>                     "
@  7, 2 SAY "  70>                        71>
72>                     "
@  8, 2 SAY "  73>                        74>
75>                     "
@  9, 2 SAY "  76>                        77>
78>                     "
@ 10, 2 SAY "  79>                        80>
81>                     "
@ 11, 2 SAY "  82>                        83>
84>                     "
@ 12, 2 SAY "  85>                        86>
87>                     "
@ 13, 2 SAY "  88>                        89>
90>                     "
@ 14, 2 SAY "  91>                        92>
93>                     "
@ 15, 2 SAY "  94>                        95>
96>                     "
@ 16, 2 SAY "  97>                        98>
99>                     "
@ 17, 0
@ 18, 0
@ 19, 0
@ 20, 2 SAY "------------------------------------------------------------------------
-----"
@ 21, 2 SAY " E to edit/add a schedule no.    L to goto last page    Q to quit this
menu "
@ 22, 2 SAY " D to delete a schedule no.      P to print schedule no.'s          "
@ 23, 2 SAY " OPTION ?"
@ 1,48 GET mgname
         CLEAR GETS

         SELECT PRIMARY
```

```
   FIND "&mkey"
   LOCATE next 56 FOR xno>=55.AND.(gno=mgno).AND.(cno=mcno).AND.(.NOT.EOF)
   DO WHILE cno=mcno.AND.gno=mgno.AND.(.NOT.EOF).AND.#<>0
      STORE int((xno-54)/3) TO row
      STORE xno-54-row*3 TO col
      IF col=0
         STORE 3 TO col
         STORE row-1 TO row
      ENDIF col=0
      @ row+2,25*(col-1)+8 SAY sno
      STORE sno TO msno
      SELECT SECOND
      FIND &msno
      IF #=0
         @ row+2,25*(col-1)+15 SAY "UnknownSchNo"
      ELSE
         @ row+2,25*(col-1)+15 SAY $(title,1,13)
      ENDIF #=0
      SELECT PRIMARY
      SKIP
   ENDDO WHILE cno=mcno.AND.gno=mgno.AND.(.NOT.eo
   STORE 'E' TO cmd

*         * edit or delect schedule number for group

CASE !(cmd)= 'E' .OR.!(cmd)= 'D'

   @ 23,18 SAY "                                                            "
   @ 23,18 SAY "which item no.? "
   IF page= 'L'
      @ 23,33 SAY " (1-54)"
      STORE 1 TO mxno
   ELSE
      @ 23,33 SAY "(55-99)"
      STORE 55 TO mxno
   ENDIF
   @ 23,40 GET mxno PICTURE "99"
   READ
   CLEAR GETS
   IF (mxno>0.AND.mxno<55.AND.page= 'L').OR.(mxno>=55.AND.mxno<=99.AND.page='N')
      STORE mkey+str(mxno,2,0) TO mkey
      FIND "&mkey"
      IF !(cmd)= 'E'
         IF #=0
            STORE "      " TO msno
            @ 23,47 SAY "adding new schedule? (y/n)"
            @ 23,74 GET mnew
            READ
            CLEAR GETS
            IF !(mnew)= 'Y'
               IF page= 'L'
                  STORE int(mxno/3) TO row
```

```
                        STORE mxno-(row*3) TO col
                     ELSE
                        STORE int((mxno-54)/3) TO row
                        STORE mxno-54-(row*3) TO col
                     ENDIF page= 'L'
                     IF col=0
                        STORE 3 TO col
                        STORE row-1 TO row
                     ENDIF col=0
                     STORE "             Enter the schedule number.
                     " TO mnotel
                     @ 23,18 GET mnotel
                     CLEAR GETS
                     @ row+2,25*(col-1)+8 GET msno PICTURE "XXXXXX"
                     READ
                     CLEAR GETS
                     SELECT secondary
                     FIND '&msno'
                     IF #=0
                        @ row+2,25*(col-1)+8 SAY "      "
                        STORE "is NOT in Project Data Base      " TO mnote3
                        @ 23,18 GET msno
                        @ 23,25 GET mnote3
                        CLEAR GETS
                        SELECT PRIMARY
                     ELSE
                        STORE $(title,1,13) TO mtitle
                        SELECT PRIMARY
                        GOTO TOP
                        LOCATE FOR (sno=msno .AND. cno=mcno)
                        IF (sno=msno .AND. cno=mcno)
                           STORE str(cno,1,0)+str(gno,2,0) TO mkey2
                           USE groups INDEX groups
                           FIND "&MKEY2"
                           STORE gname TO mgname2
                           @ row+2,25*(col-1)+8 SAY "      "
                           @ 23,20 GET msno
                           STORE "exists in group    " TO mnote3
                           @ 23,27 GET mnote3
                           @ 23,45 GET mgname2
                           CLEAR GETS
                           USE schedule INDEX schedule
                        ELSE
                           APPEND BLANK
                           REPLACE xno WITH mxno, gno WITH mgno
                           REPLACE cno WITH mcno, sno WITH msno
                           @ row+2,25*(col-1)+15 SAY mtitle
                           @ 23,18 SAY
"                                                          "
                        ENDIF (sno=msno .AND. cno=mcno)
                     ENDIF #=0
                  ELSE
```

```
      IF !(mnew) = 'N'
         @ 23,18 SAY
"                                                         "

      ENDIF !(mnew) = 'N'
   ENDIF !(mnew)= 'Y'
ELSE
   IF page= 'L'
      STORE int(mxno/3) TO row
      STORE mxno-(row*3) TO col
   ELSE
      STORE int((mxno-54)/3) TO row
      STORE mxno-54-(row*3) TO col
   ENDIF page= 'L'
   IF col=0
      STORE 3 TO col
      STORE row-1 TO row
   ENDIF col=0

   STORE sno TO msno
   STORE sno TO msno2
   @ 23,18 SAY "                           "
   @ row+2,25*(col-1)+8 GET msno2
   READ
   CLEAR GETS
   SELECT secondary
   FIND &msno2
   IF #=0
      @ row+2,25*(col-1)+8 GET msno
      CLEAR GETS
      STORE "is NOT in Project Data Base" TO mnote3
      @ 23,20 GET msno2
      @ 23,27 GET mnote3
      CLEAR GETS

   ELSE
      STORE $(title,1,13) TO mtitle
      SELECT PRIMARY
      GOTO TOP
      LOCATE FOR (sno=msno2 .AND. cno=mcno)
      IF (sno=msno2 .AND. cno=mcno)
         STORE str(cno,1,0)+str(gno,2,0) TO mkey2
         USE groups INDEX groups
         FIND "&mkey2"
         STORE gname TO mgname2
         @ row+2,25*(col-1)+8 SAY "      "
         @ row+2,25*(col-1)+8 GET msno
         CLEAR GETS
         @ 23,18 SAY "
"

         @ 23,20 GET msno2
         STORE "exists in group      " TO mnote3
         @ 23,27 GET mnote3
```

```
                        @ 23,44 GET mgname2
                        CLEAR GETS
                        USE schedule INDEX schedule
                     ELSE
                        SELECT PRIMARY
                        REPLACE sno WITH msno2
                        @ row+2,25*(col-1)+15 SAY $(title,1,13)
                     ENDIF (sno=msno2 .AND. cno=mcno)
                  ENDIF #=0
               ENDIF #=0
            ELSE
               FIND "&mkey"
               IF #=0
                  STORE "Not a Schedule No." TO mnote6
                  @ 23,50 GET mnote6
                  CLEAR GETS
               ELSE
                  @ 23,50 SAY "sure? (y/n)"
                  @ 23,65 GET mnew
                  READ
                  CLEAR GETS
                  IF !(mnew)= 'Y'
                     @ 23,18 SAY
"                                                              "

                     IF page= 'L'
                        STORE int(mxno/3) TO row
                        STORE mxno-row*3 TO col
                     ELSE
                        STORE int((mxno-54)/3) TO row
                        STORE mxno-54-row*3 TO col
                     ENDIF page= 'L'
                     IF col=0
                        STORE 3 TO col
                        STORE row-1 TO row
                     ENDIF col=0
                     STORE "WAIT...PROCESSING   " TO mnote5
                     @ row+2,25*(col-1)+8 GET mnote5
                     CLEAR GETS
                     DELETE
                     PACK
                     @ row+2,25*(col-1)+8 SAY "                     "
                  ELSE
                     IF !(mnew) = 'N'
                        @ 23,18 SAY
"                                                 "

                     ENDIF !(mnew) = 'N'
                  ENDIF !(mnew)= 'Y'
                  STORE 'E' TO cmd
               ENDIF #=0
            ENDIF !(cmd)= 'E'
         ENDIF (mxno>0.AND.mxno<55.AND.page= 'L' ).OR.
         *      * print list of projects in the group

      CASE !(cmd)= 'P'

         @ 23,18 SAY
"                                                          "
         STORE " WAIT...PREPARING REPORT FOR PRINT" TO mnote5
         @ 23,20 GET mnote5
         CLEAR GETS
         SELECT PRIMARY
         USE schedule INDEX schedule
         FIND '&mkey'
         IF EOF
            STORE " There are no Schedules Numbers in Group" TO mnote5
            @ 23,18 GET mnote5
            @ 23,40 GET mgname
            CLEAR CETS
         ELSE
            STORE p.sno TO msno
            SELECT secon
            USE pdf INDEX pdf
            @ 23,18 SAY
"                                                             "
            STORE " WAIT...PRINTING REPORT " TO mnote5
            @ 23,20 GET mnote5
            CLEAR GETS
            SET FORMAT TO PRINT
            DO WHILE .NOT. EOF.AND.(p.gno=mgno).AND.(p.cno=mcno)
@ 2,10 SAY "18th ENG BDE - Automated Construction Management System"
@ 4,5 SAY "Schedule Numbers For Project Reporting Group:"
@ 4,51 SAY mgname
@ 6,5 SAY "Sch. No.  Battalion         Title                  Location"
@ 7,5 SAY "========  =========  ======================    ================= "
               STORE 9 TO mrow2
               DO WHILE .NOT.
EOF.AND.(mrow2<62).AND.(p.cno=mcno).AND.(p.gno=mgno)
                  SELECT seco
                  FIND '&msno'
                  @ mrow2,6 SAY s.sno
                  @ mrow2,18 SAY s.bn
                  @ mrow2,28 SAY s.title
                  @ mrow2,54 SAY s.loc
                  STORE mrow2+2 TO mrow2
                  SELECT PRIMARY
                  SKIP +1
                  STORE p.sno TO msno
               ENDDO WHILE .NOT.
EOF.AND.(mrow2<62).AND.(p.cno=mcno).AND.(p.gno=mgno)
               EJECT
            ENDDO WHILE .NOT. EOF.AND.(p.gno=mgno).AND.(p.
            EJECT
            SET FORMAT TO SCREEN
            @ 23,18 SAY
"                                                               "
            STORE 'E' TO cmd
         ENDIF EOF

   ENDCASE

   @ 23,3 SAY "Option? " GET cmd PICTURE '!'
   READ
   CLEAR GETS
   STORE $(mkey,1,3) TO mkey

ENDDO WHILE !(cmd) <> 'Q'

SELECT PRIMARY
USE groups INDEX groups

* return to GROUPS.CMD
RETURN
*
*       * END SCHEDULE.CMD
*
```

```
****************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                    USA-CERL
*
*       PROGRAM: SPPDF.CMD
*
*       DESCRIPTION: Routine called by PDFMNU2.CMD to display Project Data
*                    Form on console screen.  Task displayed are results
*                    of the selection criteria, directive number, funding
*                    type, battalion, or schedule number.  Similar to
*                    PPDF.CMD.
*
*
*       CALLS: called by PDFMNU.CMD or through LISTPFD.CMD
*
****************************************************************************
ERASE
@ 1,26 SAY "TASK DATA"
@ 2, 0 SAY "PN   Number "
@ 2,15 SAY pno
@ 2,38 SAY "SCHEDULE NUMBER"
@ 2,54 SAY sno
STORE sno TO msno
@ 3, 0 SAY "Title"
@ 3, 6 SAY title
@ 3,38 SAY "Loc."
@ 3,43 SAY loc
@ 4, 0 SAY "Battalion"
@ 4,10 SAY bn
@ 4,38 SAY "Project Type"
STORE pjtype TO pj
@ 4,51 SAY mpjtype&pj
@ 5, 0 SAY "Esti. Contractor Cost"
@ 5,22 SAY ecost
@ 6, 0 SAY "Esti. Troop Const. Cost"
@ 6,24 SAY etcost
@ 7, 0 SAY "Desired Start Date (mmddyy)"
@ 7,28 SAY sdate
@ 8, 0 SAY "Desired Compl. Date (mmddyy)"
@ 8,29 SAY cdate
@ 9, 0 SAY "Customer"
@ 9, 9 SAY customer
?
? "Press any key to continue."
SET CONSOLE OFF
WAIT
SET CONSOLE ON
```

SPPDF.CMD

```
ERASE
SELECT SECONDARY
USE ps INDEX ps
FIND '&msno'
*       * allocated funds
STORE ac1 TO mac1
STORE ac2 TO mac2
STORE ac3 TO mac3
STORE ac4 TO mac4
STORE ac5 TO mac5
*       * expended funds
STORE ec1 TO mec1
STORE ec2 TO mec2
STORE ec3 TO mec3
STORE ec4 TO mec4
STORE ec5 TO mec5
SELECT PRIMARY
USE pdf INDEX pdf
FIND '&msno'
@ 1, 0 SAY "+--------------+---------------+---------------+--------------+"
@ 2, 0 SAY "|  TYPES OF    |   ESTIMATED   |   ALLOCATED   |   ACTUAL     |"
@ 3, 0 SAY "|  FUNDS       |   COST        |   FUNDS       |   EXPENDED   | "
@ 4, 0 SAY "+--------------+---------------+---------------+--------------+"
@ 5, 0 SAY "|  US TDY      |"
@ 5,17 SAY esc1
@ 5,31 SAY "|"
@ 5,33 SAY mac1
@ 5,47 SAY "|"
@ 5,50 SAY mec1
@ 5,62 SAY "|"
@ 6, 0 SAY "+--------------+---------------+---------------+--------------+"
@ 7, 0 SAY "|  US EQUIP    |"
@ 7,17 SAY esc2
@ 7,31 SAY "|"
@ 7,33 SAY mac2
@ 7,47 SAY "|"
@ 7,50 SAY mec2
@ 7,62 SAY "|"
@ 8, 0 SAY "+--------------+---------------+---------------+--------------+"
@ 9, 0 SAY "|  CSG TDY     |"
@ 9,17 SAY esc3
@ 9,31 SAY "|"
@ 9,33 SAY mac3
@ 9,47 SAY "|"
@ 9,50 SAY mec3
@ 9,62 SAY "|"
@ 10, 0 SAY "+--------------+---------------+---------------+--------------+"
@ 11, 0 SAY "|  CSG EQUIP   |"
@ 11,17 SAY esc4
@ 11,31 SAY "|"
@ 11,33 SAY mac4
@ 11,47 SAY "|"
@ 11,50 SAY mec4
@ 11,62 SAY "|"
@ 12, 0 SAY "+---------------+---------------+---------------+--------------+"
@ 13, 0 SAY "|  CSC LABOR    |"
@ 13,17 SAY esc5
@ 13,31 SAY "|"
@ 13,33 SAY mac5
@ 13,47 SAY "|"
@ 13,50 SAY mec5
@ 13,62 SAY "|"
@ 14, 0 SAY "+---------------+---------------+---------------+--------------+"
@ 15, 0 SAY "|  TOTAL        |"
STORE mac1+mac2+mac3+mac4+mac5 TO mac
STORE esc1+esc2+esc3+esc4+esc5 TO mesc
STORE mec1+mec2+mec3+mec4+mec5 TO mec
@ 15,17 SAY mesc
@ 15,31 SAY "|"
@ 15,33 SAY mac
@ 15,47 SAY "|"
@ 15,50 SAY mec
@ 15,62 SAY "|"
@ 16, 0 SAY "+---------------+---------------+---------------+--------------+"
@ 18, 0 SAY "COMMENTS:"
@ 18,10 SAY $(comments,1,40)
@ 20,10 SAY $(comments,41,29)
IF mcmd='4'
   STORE "Press any key to continue" to mnote
   @ 23,15 GET mnote
   CLEAR GETS
   SET CONSOLE OFF
   WAIT
   SET CONSOLE ON
   @ 23,0
ELSE
   STORE 'Y' TO mok
   @ 23,10 SAY "Do you wish to display next project? " GET mok PICTURE '!'
   READ
   IF mok<>"Y"
    STORE 1 TO mstop
   ENDIF
ENDIF
RETURN
*
*       * END SPPDF.CMD return to PDFMNU2.CMD or LISTPDF.CMD
*
```

```
********************************************************************************
*
*
*                        A - C - M - S  Ver. 84.0
*
*                                USA-CERL
*
*       PROGRAM: WCSRPT.CMD
*
*       DESCRIPTION: This routine generates the "Weekly Construction Status
*                    Report" which summarizes construction progress
*                    by group and gives schedule number, title, battalion,
*                    start date, completion date, percent scheduled, percent
*                    actual complete, and days off critical path.  It also
*                    computes a weighted average over the group for percent
*                    scheduled and percent complete.
*
*       CALLS: None
*
********************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
SET DELETE ON
*       * get printer/console control codes
RESTORE FROM install ADDITIVE
*       * set up default memory variables
STORE 0 TO mcno
STORE 0 TO mgno
STORE "Y" TO ALL
*
USE misc
*       * get last processing date
STORE date TO mdate
*       * present clusters for user selection
@ 1,0 SAY "               WEEKLY CONSTRUCTION STATUS REPORT"
@ 3,6 SAY "For period ending: " GET mdate PICTURE "999999"
CLEAR GETS
@ 5,0 SAY "Do you want to print out ALL the group reports for ALL of the "
@ 6,0 SAY "following clusters: (y/n) " GET ALL PICTURE '!'
*       * display clusters for user selection
USE clusters INDEX clusters
GOTO TOP
DO WHILE .NOT.EOF
   @ cno+7,2 SAY cno
   @ cno+7,3 SAY '>'
   @ cno+7,5 SAY cname
   SKIP
ENDDO WHILE .NOT.EOF
READ
```

```
CLEAR GETS

IF !(ALL)<> 'Y'
   @ cno+9,7 SAY "which cluster? " GET mcno PICTURE "9"
   @ cno+10,7 SAY "   0=EXIT"
   READ
   CLEAR GETS
   IF mcno=0
      RETURN
   ENDIF mcno=0
   STORE str(mcno,1,0) TO mkey
   FIND "&mkey"
   STORE cname TO mcna
   USE groups INDEX groups
   FIND "&mkey"
   IF #=0
      ? "                 NO Groups in Cluster"
      ? "                   Returning to Menu"
      STORE 1 TO count
      DO WHILE count<150
         STORE count+1 TO count
      ENDDO WHILE count<150
      RETURN
   ENDIF #=0
   USE schedule INDEX schedule
   COPY TO tsno FIELD sno,cno,gno FOR cno=mcno
ELSE
   USE schedules INDEX schedules
   COPY TO tsno FIELD sno,cno,gno
ENDIF !(ALL)<> 'Y'
@ 17,0
? ' Set Printer to Top Of Form'
? '  Press RETURN when ready'
WAIT
*       * clear last message
@ 16,0
@ 17,0
@ 18,0
@ 19,0
@ 20,0
@ 23,30 SAY rv+ 'PROCESSING' +rvoff
*       * routine to convert mmddyy to military date: dd MMM yy
STORE 'JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC' TO year
STORE $(mdate,1,2) TO mnth
STORE $(mdate,3,2)+ ' ' +$(year,(val(mnth)*3-2),3)+ ' ' +$(mdate,5,2) TO mil
USE tsno
GOTO TOP
*       * routine to print report
SET FORMAT TO PRINT
SET MARGIN TO 0
* set print to 12 pitch
@ 0,0 SAY p12
```

WCSRPT.CMD

```
DO WHILE .NOT.EOF
   *           * ml* are local memory variables for WCSSR
   STORE 0.00 TO mlsmh
   STORE 0.00 TO mlcws
   STORE 0.00 TO mlact
   STORE 0.00 TO mlaf
   STORE 0.00 TO mlef
   STORE str(cno,1,0)+str(gno,2,0) TO mkey
   STORE STR(cno,1,0)+STR(gno,1,0) TO mno
   SELECT SECOND
   USE groups INDEX groups
   FIND "&mkey"
   STORE gname TO mlgname
   @  1,28 SAY "WEEKLY CONSTRUCTION STATUS REPORT"
   @  3, 0 SAY ep+mlgname+epoff
   @  3,75 SAY "AS OF"
   @  3,81 SAY ep+mil+epoff
   @  6, 0 SAY "PROJECT    TITLE                 SD           CD;
           %SCH      %ACT        +/-DAYS"
   @  7, 0 SAY "-------    -----                 --           --;
           ----      ----        -------"
   STORE 9 TO row
   SELECT PRIMARY
   STORE cno TO mcno
   STORE gno TO mgno
   DO WHILE gno=mgno .AND. cno=mcno .AND. (.NOT.EOF)

      IF row>=63
         EJECT
         @  1,29 SAY "WEEKLY CONSTRUCTION STATUS REPORT"
         @  3, 0 SAY ep+mlgname+epoff
         @  3,75 SAY "AS OF"
         @  3,81 SAY ep+mil+epoff
         @  6, 0 SAY "PROJECT    TITLE                 SD           CD;
           %SCH      %ACT        +/-DAYS"
         @  7, 0 SAY "-------    -----                 --           --;
           ----      ----        -------"
         STORE 9 TO row
      ENDIF row>=63

      STORE sno TO msno
      SELECT SECOND
      USE pdf INDEX pdf
      FIND "&msno"
      @  row, 0 SAY $(msno,1,2)+'-'+$(msno,3,2)+'-'+$(msno,5,2)
      @  row,11 SAY s.title
      @  row,33 SAY $(s.sdate,1,2)+'-'+$(s.sdate,3,2)+'-'+$(s.sdate,5,2)
      @  row,46 SAY $(s.cdate,1,2)+'-'+$(s.cdate,3,2)+'-'+$(s.cdate,5,2)
      USE ps INDEX ps
      FIND "&msno"
      @  row,59 SAY s.cws USING '999.9'
      @  row,69 SAY s.actper USING '999.9'
      @  row,81 SAY s.adjust USING '9999'
      STORE s.smh+mlsmh TO mlsmh
      STORE (s.actper*s.smh)+mlact TO mlact
      STORE (s.cws*s.smh)+mlcws TO mlcws
      STORE s.ac1+s.ac2+s.ac3+s.ac4+s.ac5+mlaf TO  mlaf
      STORE s.ec1+s.ec2+s.ec3+s.ec4+s.ec5+mlef TO  mlef
      SELECT PRIMARY
      SKIP +1
      STORE row+2 TO row
   ENDDO WHILE gno=mgno .AND. cno=mcno .AND. (.NOT.EOF)
   EJECT
   SAVE TO WCSR&mno ALL LIKE ml*
ENDDO WHILE .NOT.EOF
*       * set printer back to 10 pitch
@ 0,0 SAY p10
*-------------------------------summary WCSR----------------------------
STORE '1' TO no
DO WHILE &no<=mcno
STORE '1' TO no2
STORE '0' TO no3
STORE 'A' TO ctrl1
STORE 'A' TO ctrl2
STORE 11 TO row
@  3,15 SAY "GRAF 84 WEEKLY CONSTRUCTION STATUS SUMMARY REPORT"
@  6, 8 SAY "CONSTRUCTION                                          AS OF"
@  6,67 SAY mil
@  8,35 SAY "%SCH               %ACT"
@  9,35 SAY "----               ----"
DO WHILE ctrl1='A'
   IF FILE ( 'wcsr&no&no2..mem')
     RESTORE FROM wcsr&no&no2 ADDITIVE
   ELSE
     STORE 'B' TO ctrl1
     STORE no2 TO no3
     LOOP
   ENDIF
   @ row, 8 SAY mlgname
   @ row,35 SAY mlcws/mlsmh USING '999.9'
   @ row,54 SAY mlact/mlsmh USING '999.9'
   STORE row+2 TO row
   STORE STR(&no2+1,1,0) TO no2
ENDDO
STORE '1' TO no2
STORE row+6 TO row
@ row, 8 SAY "FUNDING"
@ row+1,35 SAY "ALLOCATED          EXPENDED        %"
@ row+2,35 SAY "---------          --------        -"
STORE row+4 TO row
DO WHILE (no2<>no3)
   RESTORE FROM wcsr&no&no2 ADDITIVE
   @ row, 8 SAY mlgname
   @ row,31 SAY mlaf USING '99,999,999.99'
   @ row,49 SAY mlef USING '99,999,999.99'
   @ row,67 SAY (mlef/mlaf)*100.00 USING '999.9'
   STORE row+2 TO row
   STORE STR(&no2+1,1,0) TO no2
ENDDO
EJECT
STORE STR(&no+1,1,0) TO no
ENDDO
SET FORMAT TO SCREEN
USE
*       * delete schedule number file
DELETE FILE tsno
ERASE
RETURN
*
*       *  END WCSRPT.CMD
*
```

```
******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: WPR.CMD
*
*       DESCRIPTION: This program implements two-user Weekly Progress Report
*                    input.  The user is identified as "A" or "B" and data are
*                    stored in transaction files of the form W*A.DBF or
*                    W*B.DBF.
*
*                    The program gets the schedule number of the task and
*                    displays header information for the user to decide if it
*                    is the correct project.  After determining that the other
*                    user has not entered data for this project, the program
*                    calls WPR1.CMD to begin data entry.
*
*
*       CALLS: WPR1.CMD
*
******************************************************************************
*
ERASE
SET DELETE ON
SET COLON OFF
SET TALK OFF
*       * set up prompts and controls
STORE 'XXXXXX' TO msno
STORE 'mmddyy' TO mweek
STORE 'Y' TO mok
STORE 'Y' TO mmore
STORE ' ' TO mid
STORE 'N' TO mokid
*       * get user id : A or B
DO WHILE mokid<> 'Y'
   ERASE
   @ 13,0 SAY 'please enter USER ID (A or B) : ' GET mid PICTURE '!'
   READ
   CLEAR GETS
   IF !(mid)= 'A'
      STORE 'Y' TO mokid
      STORE 'A' TO my
      STORE 'B' TO else
   ELSE
      IF !(mid)= 'B'
         STORE 'Y' TO mokid
         STORE 'B' TO my
         STORE 'A' TO else
      ELSE
         LOOP
      ENDIF !(mid)= 'B'
   ENDIF !(mid)= 'A'

ENDDO WHILE mokid<> 'Y'
*        * get schedule number and input WPR's until user exits
DO WHILE !(mmore)<> 'N'
   @ 13,0 SAY "please enter the SCHEDULE NUMBER: " GET msno
   READ
   CLEAR GETS

   SELECT PRIMARY
   USE &my.:ps INDEX &my.:ps
   FIND '&msno'
   IF #=0
      @ 18,4
      ? "Task NOT FOUND in the project databases, check carefully."
   ELSE
      STORE date TO mdate
      USE
      @ 16,0 SAY "please enter the PERIOD END (mmddyy): ";
      GET mweek PICTURE "999999"
      READ
      CLEAR GETS

      IF mweek<=mdate
         @ 17,0 SAY "Project Status already accumulated to " +mdate
      ELSE
         USE &my.:days INDEX &my.:days
         FIND '&mweek'
         IF #=0
            @ 17,0 SAY "PERIOD END must be Friday."
            LOOP
         ELSE
            *   * check whether it is the correct task
            ERASE
            SELECT SECOND
            USE &my.:pdf INDEX &my.:pdf
            FIND '&msno'
            STORE $(title,1,18) TO mtitle
            STORE bn TO mbn
            STORE co TO mco
            STORE STR(plt,1,0) TO mplt
            STORE '&mbn'+'/'+'&mco'+'/'+'&mplt' TO munit
            @ 2,0  SAY "SECTION I: TASK IDENTIFICATION "
            @ 2,42 SAY "Period End: " GET mweek
            @ 4,0  SAY "(1) Schedule No.: " GET msno
            @ 4,42 SAY "(2) BN/CO/PLT : " GET munit
            @ 5,0  SAY "(3) Title       : " GET mtitle
            @ 5,42 SAY "(4) Location  : " GET loc
            @ 6,0  SAY "(5) Start Date  : " GET sdate
```

```
           @ 6,42 SAY "(6) Comp Date : " GET cdate
           CLEAR GETS
           @ 10,5 SAY "Is this the correct task? (y/n) " GET mok PICTURE '!'
           READ
           CLEAR GETS
           IF mok= "Y"
              @ 10,0
              @ 6,18 GET sdate
              @ 6,58 GET cdate
              READ
              CLEAR GETS
              USE
              STORE "wait... setting up for input" TO note
              @ 10,15 get note
              CLEAR GETS
              *        * see if other user has worked on this task
              USE locks&else
              SELECT PRIMARY
              USE wpr&else INDEX wpr&else
              FIND "&msno&mweek"
              IF #<>0.OR.(s.snoweek=msno+mweek)
                 @ 10,0
                 @ 19,0 SAY "The other User Station has worked on this WPR;
in the current session."
                 @ 20,0 SAY "Please work on other WPR's."
              ELSE
                 *     * set lock on this task number and week
                 SELECT SECOND
                 USE locks&my
                 REPLACE snoweek WITH msno+mweek
                 USE wpr&my INDEX wpr&my
                 FIND "&msno&mweek"
                 IF #=0
                    *     * set up for WPR data entry
                    SELECT SECONDARY
                    USE wpr&my INDEX wpr&my
                    APPEND BLANK
                    REPLACE sno WITH msno, week WITH mweek
                    USE wmdays&my INDEX wmdays&my
                    APPEND BLANK
                    REPLACE sno WITH msno, week WITH mweek
                 ENDIF #=0
                 SELECT PRIMARY
                 USE
                 SELECT SECONDARY
                 USE wpr&my INDEX wpr&my
                 FIND "&msno&mweek"

                 *********
                 DO wprl
                 *********
```

```
                  *      * clear record lock
                  USE locks&my
                  REPLACE snoweek WITH 'xxxxxxXXXXXX'
                  USE
               ENDIF #<>0.OR.(s.snoweek=msno+mweek)
            ENDIF mok= "Y"
         ENDIF #=0
      ENDIF mweek<=mdate
   ENDIF #=0

   ERASE
   STORE 'Y' TO mmore
   @ 10,5 SAY "Do you want to continue entering Weekly Reports? (y/n)" ;
   GET mmore
   READ
   CLEAR GETS
ENDDO WHILE !(mmore)<> 'N'
ERASE
RETURN
*
*       * END WPR.CMD
*
```

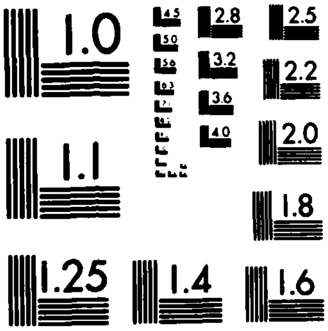
```
*****************************************************************************
*
*
*                        A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*        PROGRAM: WPR1.CMD -- Two-User Version --
*
*        DESCRIPTION: Subroutine to input data of Sections I and II of WPR.
*                     Inherited global memory variables form WPR.CMD: msno,
*                     mweek, mtitle, my, else.  Calls WPR2.CMD to complete
*                     data entry and returns to WPR.CMD.
*
*
*        CALLS: by WPR.CMD, calls WPR2.CMD
*
*****************************************************************************
*-------------------------------SECTION I------------------------------------
-
@ 10, 0 SAY "(7) Workdays Ahead (+) or Behind (-) Schedule: " GET adjust
@ 11, 0 SAY "(8) Percent Scheduled Complete From CWS: " GET cws
READ
*-----------------------------SECTION II-------------------------------------
-
USE wmdays&my INDEX wmdays&my
FIND "&msno&mweek"
ERASE
@ 1,18 SAY "SECTION II: PERSONNEL USE THIS PERIOD"
@ 2, 0 GET mtitle
@ 2,20 SAY "SCHEDULE No."
@ 2,33 GET msno
@ 2,41 SAY "PERIOD END"
@ 2,52 GET mweek
CLEAR GETS
@ 3, 0 SAY "+-------+--------------++-------------------------------------------+"
@ 4, 0 SAY "|       |   SUPPORT    ||          DIRECT CONSTRUCTION              |"
@ 5, 0 SAY "|       +--------------||--------------+------------++--------------+"
@ 6, 0 SAY "|       |     (9)      ||     (10)     |    (11)    ||     (12)     |"
@ 7, 0 SAY "|   US  | TDY ManDays  || TDY ManDays  |  ManDays   ||   ManDays    |"
@ 8, 0 SAY "|  TDY  | NOT Charged  ||  Charged     |  Worked    ||   Lost       |"
@ 9, 0 SAY "| Rate  |  To Job      ||  To Job      |  On Site   ||   Due To     |"
@ 10, 0 SAY "+-------+--------------||--------------+------------||-------+------+"
@ 11, 0 SAY "|  OFF  |"
@ 11,11 GET osupport
@ 11,23 SAY "||"
@ 11,28 GET ocharge
@ 11,39 SAY "|"
@ 11,43 GET owork
@ 11,52 SAY "|| Weath |"
@ 11,62 GET wlost
@ 11,68 SAY "|"
@ 12, 0 SAY "+-------+--------------||--------------+------------||-------+------+"
@ 13, 0 SAY "|  NCO  |"
@ 13,11 GET nsupport
@ 13,23 SAY "||"
@ 13,28 GET ncharge
@ 13,39 SAY "|"
@ 13,43 GET nwork
@ 13,52 SAY "|| Maint |"
@ 13,62 GET mlost
@ 13,68 SAY "|"
@ 14, 0 SAY "+-------+--------------||--------------+------------||-------+------+"
@ 15, 0 SAY "|  EM   |"
@ 15,11 GET esupport
@ 15,23 SAY "||"
@ 15,28 GET echarge
@ 15,39 SAY "|"
@ 15,43 GET ework
@ 15,52 SAY "|| Other |"
@ 15,62 GET olost
@ 15,68 SAY "|"
@ 16, 0 SAY "+-------+--------------++--------------+------------++-------+------+"
@ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field     ;  CTR-D: next
character ;"
@ 23, 5 SAY "CTR-W: next screen ;  CTR-S: previous character ;"
READ
CLEAR GETS
*******
DO wpr2
*******
CLEAR GETS
*       * return to WPR.CMD after WPR2.CMD
RETURN
*
*       * END WPR1.CMD
*
```

```
******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: WPR2.CMD--Two-User Version
*
*       DESCRIPTION: Final program of the WPR data input series.  Gets
*                    equipment usage data, miscellaneous costs, and
*                    manhours of work on activities in the CPM.  The program
*                    will show user any previously entered data for that
*                    week and accept changes.
*
*       CALLS: none
*
******************************************************************************
ERASE
*---------------------------------SECTION III-----------------------------------
* get equipment use data
SELECT SECOND
USE &my.:eqcost INDEX &my.:eqcost
SELECT PRIMARY
USE weq&my INDEX weq&my
@ 1, 0 SAY "SECTION III: EQUIPMENT UTILIZED THIS PERIOD   SCHEDULE NO:"
@ 1,58 GET msno
@ 1,66 SAY "PERIOD:"
@ 1,74 GET mweek
STORE 1 TO pageno
@ 3,12 SAY "+------> Enter  U for US Equip. ;  C for CSC Equip.;"
@ 4,12 SAY '|               O for Other Equip.'
@ 4,60 SAY "page " GET pageno PICTURE "9"
STORE 8 TO row
CLEAR GETS
@  5, 0 SAY "+------------+------+-------------------+-------------------+-------------------+"
@  6, 0 SAY "!Equip No.|ty.|hrs.! Equip No.|ty.|hrs.! Equip No.|ty.|hrs.! Equip No.|ty.|hrs.!"
@  7, 0 SAY "+-------------------+-------------------+-------------------+-------------------+"
@  8, 0 SAY "!      .   |   |    !      .   |   |    !      .   |   |    !      .   |   |    !"
@  9, 0 SAY "+-------------------+-------------------+-------------------+-------------------+"
@ 10, 0 SAY "!      .   |   |    !      .   |   |    !      .   |   |    !      .   |   |    !"
@ 11, 0 SAY "+-------------------+-------------------+-------------------+-------------------+"
@ 12, 0 SAY "!      .   |   |    !      .   |   |    !      .   |   |    !      .   |   |    !"
@ 13, 0 SAY "+-------------------+-------------------+-------------------+-------------------+"
@ 14, 0 SAY "!      .   |   |    !      .   |   |    !      .   |   |    !      .   |   |    !"
@ 15, 0 SAY "+-------------------+-------------------+-------------------+-------------------+"
@ 18, 0 SAY "Enter as many entries as you want; NOTE that only the row located by "
@ 19, 0 SAY "the blinking curser can be entered or changed."
@ 20,0 SAY "To delete a entry, leave the  EQUIP NO. of that entry to be '0' ."
@ 21,0 SAY  "NOTE: USE CTRL-W TO SCROLL THROUGH OLD ENTRIES"
FIND &msno&mweek
```

```
* loop to let user change any data previously entered for this week
DO WHILE (.NOT.EOF).AND.sno=msno.AND.week=mweek.AND.(#<>0)
   @ 4,60 SAY "page " GET pageno PICTURE "9"
   CLEAR GETS
   @ row,2 GET eno1
   @ row,7 GET atchno1
   @ row,12 GET type1 PICTURE '!'
   @ row,15 GET hr1
   @ row,22 GET eno2
   @ row,27 GET atchno2
   @ row,32 GET type2 PICTURE '!'
   @ row,35 GET hr2
   @ row,42 GET eno3
   @ row,47 GET atchno3
   @ row,52 GET type3 PICTURE '!'
   @ row,55 GET hr3
   @ row,62 GET eno4
   @ row,67 GET atchno4
   @ row,72 GET type4 PICTURE '!'
   @ row,75 GET hr4
   READ
   CLEAR GETS
   STORE "1" TO no
   * routine to verify that equipment is in eqcost.dbf
   DO WHILE no<= "4"
      IF eno&no=0
         REPLACE atchno&no WITH 0,type&no WITH " " ,hr&no WITH 0
         STORE str(&no+1,1,0) TO no
         LOOP
      ENDIF eno&no=0
      STORE STR(p.eno&no,4,0)+STR(p.atchno&no,2,0) TO meq
      SELECT SECONDARY
      FIND '&meq'
      DO WHILE #=0.AND.eno&no<>0
         SELECT PRIMARY
         @ 16,0 SAY " "+CHR(7)
         ? "        Unknown EQUIPMENT CODE,    try again.  "
         @ row,(&no-1)*20+2 GET eno&no
         @ row,(&no-1)*20+7 GET atchno&no
         @ row,(&no-1)*20+12 GET type&no
         @ row,(&no-1)*20+15 GET hr&no
         READ
         CLEAR GETS
         STORE STR(p.eno&no,4,0)+STR(p.atchno&no,2,0) TO meq
         SELECT SECONDARY
         FIND '&meq'
      ENDDO WHILE #=0.AND.eno&no<>0
      SELECT PRIMARY
      IF eno&no=0
         REPLACE atchno&no WITH 0,type&no WITH " " ,hr&no WITH 0
         STORE str(&no+1,1,0) TO no
         LOOP
      ENDIF eno&no=0
      * check for illegal type
      DO WHILE !(type&no)<> 'U'.AND.!(type&no)<> 'O' .AND.!(type&no)<>'C'
         @ 16,0 SAY " "
         ? '     Illegal type, must be "U","C" or "O" '
         @ row,(&no-1)*20+12 GET type&no
         READ
         CLEAR GETS
      ENDDO WHILE
      @ 17,0
      STORE str(&no+1,1,0) TO no
   ENDDO WHILE no<= "4"
   STORE row+2 TO row
   IF row>14
      STORE 8 TO row
      STORE pageno+1 TO pageno
   ENDIF row>14
   SKIP 1
ENDDO WHILE (.NOT.EOF).AND.sno=msno.AND.week=mweek.AND.(#<>0)
* now get new equipment data
STORE "Y" TO amore
@ 16,0 SAY "want to add NEW entries of equipment utilized? (y/n)" GET amore
READ
CLEAR GETS
@ 16,0
DO WHILE !(amore)= 'Y'
   SELECT PRIMARY
   APPEND BLANK
   REPLACE sno WITH msno,week WITH mweek,type1 WITH "U" ,type2 WITH "U" ,type3 WITH "U" ,
      type4 WITH "U"
   @ 4,60 SAY "page " GET pageno PICTURE "9"
   CLEAR GETS
   @ row,2 GET eno1
   @ row,7 GET atchno1
   @ row,12 GET type1 PICTURE '!'
   @ row,15 GET hr1
   @ row,22 GET eno2
   @ row,27 GET atchno2
   @ row,32 GET type2 PICTURE '!'
   @ row,35 GET hr2
   @ row,42 GET eno3
   @ row,47 GET atchno3
   @ row,52 GET type3 PICTURE '!'
   @ row,55 GET hr3
   @ row,62 GET eno4
   @ row,67 GET atchno4
   @ row,72 GET type4 PICTURE '!'
   @ row,75 GET hr4
   READ
   CLEAR GETS
   STORE "1" TO no
   * routine to verify that equipment is in eqcost.dbf
   DO WHILE no<= "4"
      IF eno&no=0
         REPLACE atchno&no WITH 0,type&no WITH " " ,hr&no WITH 0
         STORE str(&no+1,1,0) TO no
         LOOP
      ENDIF eno&no=0
      STORE STR(p.eno&no,4,0)+STR(p.atchno&no,2,0) TO meq
      SELECT SECONDARY
      FIND '&meq'
      DO WHILE #=0.AND.eno&no<>0
         SELECT PRIMARY
         @ 16,0 SAY " "+CHR(7)
         ? "        Unknown EQUIPMENT CODE,    try again.   "
         @ row,(&no-1)*20+2 GET eno&no
         @ row,(&no-1)*20+7 GET atchno&no
         @ row,(&no-1)*20+12 GET type&no
         @ row,(&no-1)*20+15 GET hr&no
         READ
         CLEAR GETS
         STORE STR(p.eno&no,4,0)+STR(p.atchno&no,2,0) TO meq
         SELECT SECONDARY
         FIND '&meq'
      ENDDO WHILE #=0.AND.eno&no<>0
      SELECT PRIMARY
      IF eno&no=0
         REPLACE atchno&no WITH 0,type&no WITH " " ,hr&no WITH 0
         STORE str(&no+1,1,0) TO no
         LOOP
      ENDIF eno&no=0
      DO WHILE !(type&no)<> 'U' .AND.!(type&no)<> 'O'.AND.!(type&no)<> 'C'
         @ 16,0 SAY " "
         ? '      Illegal type, must be  "U", "C" or "O" '
         @ row,(&no-1)*20+12 GET type&no
         READ
         CLEAR GETS
      ENDDO WHILE
      @ 17,0
      STORE str(&no+1,1,0) TO no
   ENDDO WHILE no<= "4"
   STORE row+2 TO row
   @ 16,10 SAY "continue? (y/n)" GET amore PICTURE '!'
   READ
   CLEAR GETS
   IF row>14
      STORE 8 TO row
      STORE pageno+1 TO pageno
   ENDIF row>14
ENDDO WHILE !(amore)= 'Y'
*---------------------------------SECTION IV------------------------------------
* get other costs
ERASE
@ 2, 18 SAY "SECTION IV: OTHER COST(S) THIS PERIOD"
@ 4, 0 SAY "SCHEDULE NO.: " GET msno
@ 4,26 SAY "TITLE: " GET mtitle
@ 4,54 SAY "PERIOD END :" GET mweek
CLEAR GETS
SELECT PRIMARY
USE wpr&my INDEX wpr&my
FIND '&msno&mweek'
@  7, 0 SAY "(18) CSC Labor     : " GET csglbr
@  7,37 SAY "(21) Rental Fees   : " GET rental
@  9, 0 SAY "(19) CSC TDY       : " GET csgtdy
@  9,37 SAY "(22) Transportation: " GET trans
@ 11, 0 SAY "(20) CSC Equipment : " GET csgeq
@ 11,37 SAY "(23) Other US TDY  : " GET other
READ
CLEAR GETS
ERASE
*--------------------------------SECTION V------------------------------------
* get manhours of work by activities: i-j's in the CPM
@ 2,18 SAY "SECTION V: CONSTRUCTIVE WORK BY ACTIVITY"
@ 4, 0 SAY "SCHEDULE NO.: " GET msno
@ 4,26 SAY "TITLE: " GET mtitle
@ 4,54 SAY "PERIOD END :" GET mweek
@  8, 1 SAY "+--------+--------+-----------------+------------+----------+"
@  9, 1 SAY "¢       (25)      ¢       (26)      ¢    (28)    ¢   (31)   ¢"
@ 10, 1 SAY "¢   I    ¢    J   ¢     SUBTASKS    ¢ ACTUAL     ¢ PHY  %   ¢"
@ 11, 1 SAY "¢        ¢        ¢                 ¢ MAN HOURS  ¢ COMPLETE ¢"
@ 12, 1 SAY "+--------+--------+-----------------+------------+----------+"
STORE 'wait..one' TO note
@ 21, 5 GET note
CLEAR GETS
SELECT SECOND
USE &my.:act INDEX &my.:act
SELECT PRIMARY
USE wact&my INDEX wact&my
FIND '&msno&mweek'
@ 21,0
* loop to let user change previously enterd data
DO WHILE (.NOT.EOF).AND.(p.sno=msno).AND.(mweek=week).AND.(#<>0)
STORE "NOTE: To DELETE and unwanted I-J entry set Hours and Per.Comp. to 0" TO mnote
@ 7,1 GET mnote
CLEAR GETS
    @ 13, 1 SAY "¢        ¢        ¢                 ¢            ¢          ¢"
    @ 14, 1 SAY "+--------+--------+-----------------+------------+----------+"
    @ 15, 1 SAY "¢        ¢        ¢                 ¢            ¢          ¢"
    @ 16, 1 SAY "+--------+--------+-----------------+------------+----------+"
    @ 17, 1 SAY "¢        ¢        ¢                 ¢            ¢          ¢"
    @ 18, 1 SAY "+--------+--------+-----------------+------------+----------+"
    @ 19, 1 SAY "¢        ¢        ¢                 ¢            ¢          ¢"
    @ 20, 1 SAY "+--------+--------+-----------------+------------+----------+"
    @ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field  ;"
    @ 23, 0 SAY "     CTR-W: next screen ;  CTR-D: next character  ;  CTR-S: next character  ;'
    STORE 1 TO nu
    STORE 13 TO row
    DO WHILE (nu<5).AND.(.NOT.EOF).AND.(p.sno=msno).AND.(mweek=week).AND.(#<>0)
      @ row, 3 GET p.i
      @ row,12 GET p.j
        STORE msno+STR(p.i,4,0)+STR(p.j,4,0) TO mkey
        SELECT SECONDARY
        FIND '&mkey'
        @ row,21 GET s.des
        CLEAR GETS
        SELECT PRIMARY
        @ row,39 GET p.amh
        @ row,52 GET p.per
        READ
        CLEAR GETS
        STORE nu+1 TO nu
        STORE row+2 TO row
        SKIP +1
    ENDDO while (nu<5)
ENDDO
@ 7,0
STORE 'NOTE: Enter 0 for Start Node to STOP inputting' to note
@  7,10 GET note
CLEAR GETS
STORE "Y" TO mmore
DO WHILE !(mmore) <> "N"
      @ 21, 5
      STORE "Y" TO mmore
      STORE "want to add new i-j's ?(y/n)" TO note
      @ 21, 5
      @ 21, 5 GET note
      CLEAR GETS
      @ 21,34 GET mmore
      READ
      @ 21, 5
      CLEAR GETS
      IF !(mmore)<>"Y"
         STORE "N" TO mmore
         LOOP
      ENDIF
    @ 13, 1 SAY "¢        ¢        ¢                 ¢            ¢            ¢"
    @ 14, 1 SAY "+--------+--------+-----------------+------------+------------+"
    @ 15, 1 SAY "¢        ¢        ¢                 ¢            ¢            ¢"
    @ 16, 1 SAY "+--------+--------+-----------------+------------+------------+"
    @ 17, 1 SAY "¢        ¢        ¢                 ¢            ¢            ¢"
    @ 18, 1 SAY "+--------+--------+-----------------+------------+------------+"
    @ 19, 1 SAY "¢        ¢        ¢                 ¢            ¢            ¢"
    @ 20, 1 SAY "+--------+--------+-----------------+------------+------------+"
    @ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field  ;"
    @ 23, 0 SAY "     CTR-W: next screen ;  CTR-D: next character  ;  CTR-S: next character  ;"
    STORE 1 TO nu
    STORE 13 TO row
    DO WHILE (nu<5)
```

AD-A143 032 AUTOMATED CONSTRUCTION MANAGEMENT SYSTEM (ACMS) VOLUME 2 PROGRAM DOCUMENTATION(U) CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL C E HERRING JUN 84 3/3

UNCLASSIFIED CERL-TR-P-158-VOL-2 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
       STORE 0 TO mi
       @ row, 3 GET mi PICTURE '9999'
       READ
       @ 21,0
       CLEAR GETS
       IF mi=0
          STORE 5 TO nu
          STORE 'N' TO mmore
          LOOP
       ENDIF
       STORE 0 TO mj
       @ row,12 GET mj PICTURE '9999'
       READ
       CLEAR GETS
       STORE STR(mi,4,0)+STR(mj,4,0) TO mij
       STORE msno+mweek+mij TO mwkey
       STORE msno+mij TO mkey
       SELECT SECONDARY
       FIND "&mkey"
       IF #=0
          * activity not in data base
          STORE 'I-J NOT FOUND'+CHR(7) TO note
          @ 21,5 GET note
          CLEAR GETS
          LOOP
       ENDIF
       SELECT PRIMARY
       FIND '&mwkey'
       IF #=0
         APPEND BLANK
         REPLACE i WITH mi, j WITH mj, sno WITH msno, week WITH mweek
       ENDIF
       @ row,21 GET des
       CLEAR GETS
       @ row,39 GET p.amh
       @ row,52 GET p.per
       READ
       CLEAR GETS
       STORE nu+1 TO nu
       STORE row+2 TO row
    ENDDO while (nu<5)
ENDDO while !(mmore)<>"N"
ERASE
CLEAR GETS
RETURN
*
*  END WPR2.CMD
*
```

```
*************************************************************************************
*
*                             A - C - M - S  Ver. 84.0
*
*                                       USA-CERL
*
*        PROGRAM: PS.CMD -- Single-User Version --
*
*        DESCRIPTION: This program processes the Weekly Progress Reports.
*                     The weekly input transaction files W*.DBF are used
*                     to update the master files.  Only tasks that have WPR
*                     data in WPR.DBF are involved in this program.
*
*        CALLS: none
*
*************************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
SET DELETE ON
*        * get printer/console control codes
RESTORE FROM install ADDITIVE
* * * * * * * * * * * * * * * MAIN PROCESSING ROUTINE * * * * * * * * * * * * *
ERASE
USE misc
*        * get graff tdy rates
STORE goff TO goff
STORE gnco TO gnco
STORE gem TO gem
*        * get date of last processing
STORE date TO mdate
@ 1,0 SAY '                    PROCESS WEEKLY PROGRESS REPORTS'
USE days INDEX days
*        * days contains fridays
FIND '&mdate'
*        * get this week
SKIP +1
STORE day TO mdate
STORE mdate TO mdatel
STORE "N" TO msure
*        * give user option to process for any period
DO WHILE !(msure)<> "Y"
   @ 4,0 SAY "Enter date of the period end(Friday): " GET mdate
   @ 5,0 SAY '     or press RETURN if correct'
   READ
   CLEAR GETS
   IF mdate<>mdatel
      FIND '&mdate'
      IF #=0
         @ 6,0 SAY 'Period Ending date must be Friday.  Enter again.'
         STORE mdatel TO mdate
      ELSE
```

```
         @ 5,0
         @ 6,0
         @ 6,0 SAY "Sure? (y/n) " GET msure PICTURE '!'
         READ
         CLEAR GETS
      ENDIF #=0
   ELSE
      STORE "Y" TO msure
   ENDIF mdate<>mdatel
ENDDO WHILE !(msure)<> "Y"

USE misc
*        * store processing date for next time
REPLACE date WITH mdate
SELECT PRIMARY
USE ps INDEX ps
SELECT SECOND
USE wpr INDEX wpr
@ 12,0
@ 12,30 SAY rv+'PROCESSING'+rvoff

DO WHILE .not.EOF
   STORE s.sno TO msno
   SELECT PRIMARY
   FIND "&msno"
   IF p.date>=mdate
      SELECT SECOND
      FIND "&msno"
      DO WHILE (s.sno=msno).AND.(.not.EOF)
                 SKIP +1
      ENDDO WHILE (s.sno=msno).AND.(.not.EOF)
      LOOP
   ELSE
      SELECT SECOND
      FIND "&msno"
      * skip previously processsed wprs
      DO WHILE (s.sno=msno).AND.(week<=p.date).AND.(.not.EOF)
         SKIP +1
      ENDDO WHILE (s.sno=msno).AND.(week<=p.date).AN
      * * this loop processes all wpr ' s up to the desired date.
      DO WHILE (s.sno=msno).AND.(s.week>p.date).AND.(s.week<=mdate)
         @ 14,20 SAY "        Schedule No.:" GET msno
         CLEAR GETS
         STORE sno+week TO msnoweek
         SELECT PRIMARY
         REPLACE p.date WITH s.week
         REPLACE p.cws WITH s.cws
         REPLACE p.adjust WITH s.adjust
         REPLACE ec5 WITH ec5+s.csglbr
         REPLACE ec4 WITH ec4+s.csgeq
         REPLACE ec3 WITH ec3+s.csgtdy
         REPLACE ecl WITH ecl+s.other
REPLACE ec2 WITH ec2+s.rental+s.trans
*       * processing mandays data
SELECT SECOND
USE pdf INDEX pdf
FIND "&msno"
IF !(graf)= "N"
   *            * for non-graf projects, find the particular tdy rates
   STORE OFF TO ngoff
   STORE nco TO ngnco
   STORE em TO ngem
   USE wmandays INDEX wmandays
   FIND "&msnoweek"
   SELECT PRIMARY
   *            * primary is ps
   REPLACE ecl WITH ecl+s.ocharge*ngoff+s.ncharge*ngnco+s.echarge*ngem
ELSE
   USE wmandays INDEX wmandays
   FIND "&msnoweek"
   SELECT PRIMARY
   REPLACE ecl WITH ecl+s.ocharge*goff+s.ncharge*gnco+s.echarge*gem
ENDIF !(graf)= "N"
*-----------------------------
*       * processing equip data
*-----------------------------
SELECT PRIMARY
USE eqcost INDEX eqcost
SELECT SECOND
USE weq INDEX weq
FIND "&msnoweek"
*       * use useqcost to store us equip cost.
STORE 0 TO useqcost
STORE 0 TO csceqcost
*       * loop through all the weq records of this wpr
DO WHILE (s.sno+s.week=msnoweek).AND.(.NOT.EOF).AND.#<>0
   STORE "1" TO no
   *            * looping through all 4 equip items in one weq record
   DO WHILE no<= "4"
    IF s.eno&no<>0
      IF s.type&no= "O"
         STORE str(&no+1,1,0) TO no
         LOOP
      ENDIF
      STORE str(s.eno&no,4,0)+str(s.atchno&no,2,0) TO mkey
      SELECT PRIMARY
      FIND "&mkey"
      IF s.type&no='U'
        STORE p.omcost*s.hr&no +useqcost TO useqcost
      ELSE
        STORE p.omcost*s.hr&no+csceqcost TO csceqcost
      ENDIF
    ENDIF s.eno&no<>0
    STORE str(&no+1,1,0) TO no
   ENDDO WHILE no<= "4"
   SELECT SECOND
   SKIP +1
ENDDO WHILE (sno+week=msnoweek).AND.(.not.EOF)
*------------------------------
*      * processing activity data
*------------------------------
SELECT SECOND
USE wact INDEX wact
FIND '&msnoweek'
SELECT PRIMARY
USE act INDEX act
STORE 0.0 TO mactper
STORE 0 TO mtamh
SUM p.smh TO mtsmh FOR p.sno=msno
FIND '&msno'
DO WHILE (p.sno=msno).AND.(s.sno+s.week=msnoweek).AND.(.NOT.EOF).AND.(#<>0)
   STORE msnoweek+STR(p.i,4,0)+STR(p.j,4,0) TO mij
   SELECT SECOND
   FIND '&mij'
   IF #<>0
       SELECT PRIMARY
       REPLACE p.amh WITH p.amh+s.amh
       REPLACE p.per WITH s.per
   ELSE
       SELECT PRIMARY
   ENDIF
   STORE p.amh+mtamh TO mtamh
   STORE (p.smh/mtsmh)*p.per+mactper TO mactper
   SKIP +1
ENDDO
*---------------------------
*      * update mandays master
*---------------------------
SELECT SECOND
USE wmandays INDEX wmandays
FIND "&msnoweek"
SELECT PRIMARY
USE mandays INDEX mandays
FIND "&msno"
REPLACE osupport WITH osupport+s.osupport
REPLACE nsupport WITH nsupport+s.nsupport
REPLACE esupport WITH esupport+s.esupport
REPLACE ocharge  WITH ocharge+s.ocharge
REPLACE owork    WITH owork+s.owork
REPLACE ncharge  WITH ncharge+s.ncharge
REPLACE nwork    WITH nwork+s.nwork
REPLACE echarge  WITH echarge+s.echarge
REPLACE ework    WITH ework+s.ework
REPLACE wlost    WITH wlost+s.wlost
REPLACE mlost    WITH mlost+s.mlost
REPLACE olost    WITH olost+s.olost
```

```
         *----------------------------------------------------
         *      * accumulate useq cost,csceq cost and per comp
         *----------------------------------------------------
         SELECT PRIMARY
         USE ps INDEX ps
         FIND "&msno"
         REPLACE ec2 WITH ec2+useqcost
         REPLACE ec4 WITH ec4+csceqcost
         REPLACE actper WITH mactper
         REPLACE amh WITH mtamh
         *      * get next wpr to process
         SELECT SECOND
         USE wpr INDEX wpr
         FIND "&msno"
         DO WHILE (sno=msno).AND.(week<=p.date).AND.(.not.EOF)
            SKIP 1
         ENDDO WHILE (sno=msno).AND.(week<=p.date).AND.
      ENDDO WHILE (s.sno=msno).AND.(s.week>p.date).a
   ENDIF p.date>=mdate

ENDDO WHILE .not.EOF

ERASE
RETURN
*
*       * END PS.CMD
*
```

```
*******************************************************************************
*
*
*                            A - C - M - S  Ver. 84.0
*
*                                   USA-CERL
*
*       PROGRAM: BACKUP.CMD
*
*       DESCRIPTION: Data base backup program to clear processed data from
*                    weekly temporary data bases.  All weekly input data base
*                    file names begin with "W".  Backs off data contained in
*                    WPR.DBF, WMANDAYS.DBF, WACT.DBF, and WEQ.DBF.  Copies all
*                    data entered on or after the last processing date.
*
*       CALLS: none: quits to A:SUBMIT PIPBAK
*
*******************************************************************************
ERASE
SET TALK OFF
SET COLON OFF
@ 1,0
? '                          WEEKLY DATA BASE BACKUP'
CLEAR GETS

USE misc
STORE date TO mdate
USE
*       * all created .dbf files have month and day in file name
*       * example: WPR0708.DBF <= WPR&mmdd.DBF
STORE $(mdate,1,4) TO mmdd

@ 5,0
? '                          Backing off CERL data '
USE wpr INDEX wpr
COPY TO wpr&mmdd FOR week<=mdate
COPY TO tt FOR week>mdate
USE tt
COPY TO wpr
USE wpr
INDEX ON sno+week TO wpr

? '                          Backing off old mandays data'
USE wmandays INDEX wmandays
COPY TO wman&mmdd FOR week<=mdate
COPY TO tt FOR week>mdate
USE tt
COPY TO wmandays
USE wmandays
INDEX ON sno+week TO wmandays
```

BACKUP.CMD

```
? '                          Backing off old CPM data'
USE wact INDEX wact
COPY TO wact&mmdd FOR week<=mdate
COPY TO tt FOR week>mdate
USE tt
COPY TO wact
USE wact
INDEX ON sno+week+str(i,4,0)+str(j,4,0) TO wact

? '                          Backing off old equipment data'
USE weq INDEX weq
COPY TO weq&mmdd FOR week<=mdate
COPY TO tt FOR week>mdate
USE eq INDEX eq
append from weq&mmdd
USE tt
COPY TO weq
USE weq
INDEX ON sno+week TO weq
USE
DELETE FILE tt
ERASE
@ 1,0
?
? "                   Insert Diskette in Drive to make CERL Data Disk"
?
? "                          Press RETRUN when ready to copy"
SET CONSOLE OFF
WAIT
SET CONSOLE ON
?
? "                                 WAIT...COPYING"
*       * c is floppy on TeleVideo TS-806
SET DEFA TO C:
RESET C:
SET DEFA TO A:
USE wpr&mmdd
COPY TO c:wpr&mmdd
USE wman&mmdd
COPY TO c:wman&mmdd
USE weq&mmdd
COPY TO c:weq&mmdd
USE  wact&mmdd
COPY TO c:wact&mmdd
USE
*       * make sure it is copied before delete
IF FILE ('c:wpr&mmdd')
DELETE FILE a:wpr&mmdd
ENDIF
IF FILE ('c:wman&mmdd')
DELETE FILE a:wman&mmdd
ENDIF
```

```
IF FILE ('c:weq&mmdd')
DELETE FILE a:weq&mmdd
ENDIF
IF FILE ('c:wact&mmdd')
DELETE FILE a:wact&mmdd
ENDIF
*       * get this weeks ps.dbf for analysis
USE
USE ps
COPY TO c:ps&mmdd..dbf
USE
RETURN
*
*       * END BACKUP.CMD
*
```

```
*******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                 USA-CERL
*
*       PROGRAM: WPR.CMD -- Single-User Version --
*
*       DESCRIPTION: This program implements single user Weekly Progress Report
*                    input.  The program gets the task schedule number and
*                    displays header information for the user to decide if it
*                    is the correct task.  The program calls WPR1.CMD to begin
*                    data entry.
*
*       CALLS: WPR1.CMD
*
*******************************************************************************
*
ERASE
SET DELETE ON
SET COLON OFF
SET TALK OFF
*       * set up prompts and controls
STORE 'XXXXXX' TO msno
STORE 'mmddyy' TO mweek
STORE 'Y' TO mok
STORE 'Y' TO mmore
STORE 'N' TO mokid
*       * get schedule numbers and input WPR's until user exits
DO WHILE !(mmore)<
   @ 13,0 SAY "please enter the SCHEDULE NUMBER: " GET msno
   READ
   CLEAR GETS

   SELECT PRIMARY
   USE ps INDEX ps
   FIND '&msno'
   IF #=0
      @ 18,4
      ? "Task NOT FOUND in the project databases, check carefully."
   ELSE
      STORE date TO mdate
      USE
      @ 16,0 SAY "please enter the PERIOD END (mmddyy): " ;
      GET mweek PICTURE "999999"
      READ
      CLEAR GETS

      IF mweek<=mdate
         @ 17,0 SAY "Project Status already accumulated to " +mdate
```

```
ELSE
   USE days INDEX days
   FIND '&mweek'
   IF #=0
      @ 17,0 SAY "PERIOD END must be Friday."
      LOOP
   ELSE
      *   * check whether it is the correct task
      ERASE
      SELECT SECOND
      USE pdf INDEX pdf
      FIND '&msno'
      STORE $(title,1,18) TO mtitle
      STORE bn TO mbn
      STORE co TO mco
      STORE STR(plt,1,0) TO mplt
      STORE '&mbn'+'/'+'&mco'+'/'+'&mplt' TO munit
      @ 2,0  SAY "SECTION I: TASK IDENTIFICATION "
      @ 2,42 SAY "Period End: " GET mweek
      @ 4,0  SAY "(1) Schedule No.: " GET msno
      @ 4,42 SAY "(2) BN/CO/PLT : " GET munit
      @ 5,0  SAY "(3) Title       : " GET mtitle
      @ 5,42 SAY "(4) Location  : " GET loc
      @ 6,0  SAY "(5) Start Date  : " GET sdate
      @ 6,42 SAY "(6) Comp Date : " GET cdate
      CLEAR GETS
      @ 10,5 SAY "Is this the correct task? (y/n) ";
      GET mok PICTURE '!'
      READ
      CLEAR GETS
      IF mok= "Y"
         @ 10,0
         @ 6,18 GET sdate
         @ 6,58 GET cdate
         READ
         CLEAR GETS
         USE
         STORE "wait... setting up for input" TO note
         @ 10,15 get note
         CLEAR GETS
            USE wpr INDEX wpr
            FIND "&msno&mweek"
            IF #=0
               * set up for WPR data entry
               SELECT SECONDARY
               USE wpr INDEX wpr
               APPEND BLANK
               REPLACE sno WITH msno, week WITH mweek
               USE wmandays INDEX wmandays
               APPEND BLANK
               REPLACE sno WITH msno, week WITH mweek
            ENDIF #=0
```

```
                  SELECT PRIMARY
                  USE
                  SELECT SECONDARY
                  USE wpr INDEX wpr
                  FIND "&msno&mweek"

                  *******
                  DO wpr1
                  *******

               ENDIF #<>0.OR.(s.snoweek=msno+mweek)
            ENDIF mok= "Y"
         ENDIF #=0
      ENDIF mweek<=mdate
   ENDIF #=0

   ERASE
   STORE 'Y' TO mmore
   @ 10,5 SAY "Do you want to continue entering Weekly Reports? (y/n)" ;
   GET mmore
   READ
   CLEAR GETS
ENDDO WHILE !(mmore)<> 'N'
USE
ERASE
RETURN
*
*       * END WPR.CMD
*
```

```
******************************************************************************
*
*
*                          A - C - M - S  Ver. 84.0
*
*                                  USA-CERL
*
*       PROGRAM: WPR1.CMD -- Single-User Version --
*
*       DESCRIPTION: Subroutine to input data of Sections I and II of WPR.
*                    Inherited global memory variables form WPR.CMD: msno,
*                    mweek, mtitle, my, else.  Calls WPR2.CMD to complete
*                    data entry and returns to WPR.CMD.
*
*
*       CALLS: by WPR.CMD, calls WPR2.CMD
*
*******************************************************************************
*---------------------------------SECTION I-------------------------------------
@ 10, 0 SAY "(7) Workdays Ahead (+) or Behind (-) Schedule: " GET adjust
@ 11, 0 SAY "(8) Percent Scheduled Complete From CWS: " GET cws
READ
*-------------------------------SECTION II-------------------------------------
USE wmandays INDEX wmandays
FIND "&msno&mweek"
ERASE
@ 1,18 SAY "SECTION II: PERSONNEL USE THIS PERIOD"
@ 2, 0 GET mtitle
@ 2,20 SAY "SCHEDULE No."
@ 2,33 GET msno
@ 2,41 SAY "PERIOD END"
@ 2,52 GET mweek
CLEAR GETS
@ 3, 0 SAY "+-------+--------------++-------------------------------------------+"
@ 4, 0 SAY "|       |   SUPPORT    ||           DIRECT CONSTRUCTION             |"
@ 5, 0 SAY "|       +--------------||--------------+------------++--------------+"
@ 6, 0 SAY "|       |     (9)      ||     (10)     |    (11)    ||     (12)     |"
@ 7, 0 SAY "|  US   | TDY ManDays  ||  TDY ManDays |  ManDays   ||    ManDays   |"
@ 8, 0 SAY "|  TDY  | NOT Charged  ||   Charged    |  Worked    ||    Lost      |"
@ 9, 0 SAY "|  Rate |  To Job      ||   To Job     |  On Site   ||    Due To    |"
@ 10, 0 SAY "+-------+--------------||--------------+------------||-------+------+"
@ 11, 0 SAY "|  OFF  |"
@ 11,11 GET osupport
@ 11,23 SAY "||"
@ 11,28 GET ocharge
@ 11,39 SAY "|"
@ 11,43 GET owork
@ 11,52 SAY "|| Weath |"
@ 11,62 GET wlost
@ 11,68 SAY "|"
@ 12, 0 SAY "+-------+--------------|'      --------+------------||-------+------+"
@ 13, 0 SAY "|  NCO   |"
@ 13,11 GET nsupport
@ 13,23 SAY "||"
@ 13,28 GET ncharge
@ 13,39 SAY "|"
@ 13,43 GET nwork
@ 13,52 SAY "|| Maint |"
@ 13,62 GET mlost
@ 13,68 SAY "|"
@ 14, 0 SAY "+-------+-------------||-------------+-------------||-------+------+"
@ 15, 0 SAY "|  EM    |"
@ 15,11 GET esupport
@ 15,23 SAY "||"
@ 15,28 GET echarge
@ 15,39 SAY "|"
@ 15,43 GET ework
@ 15,52 SAY "|| Other |"
@ 15,62 GET olost
@ 15,68 SAY "|"
@ 16, 0 SAY "+-------+-------------++-------------+-------------++-------+------+"
@ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field    ;  CTR-D: next
character ;"
@ 23, 5 SAY "CTR-W: next screen ;  CTR-S: previous character ;"
READ
CLEAR GETS

*********
DO wpr2
*********

CLEAR GETS
RETURN
* return to WPR.CMD after WPR2.CMD
*
*       * END WPR1.CMD
*
```

```
********************************************************************************
*
*
*                             A - C - M - S  Ver. 84.0
*
*                                       USA-CERL
*
*       PROGRAM: WPR2.CMD -- Single-User Version --
*
*       DESCRIPTION: Final program of the WPR data input series.  Gets
*                    equipment usage data, miscellaneous costs, and
*                    manhours of work on subtasks in the CPM.  The program
*                    will show user any previously entered data for that
*                    week and accept changes.
*
*       CALLS: none
*
********************************************************************************
ERASE
*----------------------------------SECTION III---------------------------------
--
*       * get equipment use data
SELECT SECOND
USE eqcost INDEX eqcost
SELECT PRIMARY
USE weq INDEX weq
@ 1, 0 SAY "SECTION III: EQUIPMENT UTILIZED THIS PERIOD   SCHEDULE NO:"
@ 1,58 GET msno
@ 1,66 SAY "PERIOD:"
@ 1,74 GET mweek
STORE 1 TO pageno
@ 3,12 SAY "+------> Enter  U for US Equip. ;  C for CSC Equip.;"
@ 4,12 SAY '|               O for Other Equip.'
@ 4,60 SAY "page " GET pageno PICTURE "9"
STORE 8 TO row
CLEAR GETS
@  5, 0 SAY "+-----------+-------+-------------------+-------------------+-----
--------------+"
@  6, 0 SAY "!Equip No.|ty.|hrs.! Equip No.|ty.|hrs.! Equip No.|ty.|hrs.!
Equip No.|ty.|hrs.!"
@  7, 0 SAY "+-------------------+-------------------+-------------------+-----
--------------+"
@  8, 0 SAY "!     .   |   |    !      .   |   |    !      .   |   |
!     .   |   |    !"
@  9, 0 SAY "+-------------------+-------------------+-------------------+-----
--------------+"
@ 10, 0 SAY "!     .   |   |    !      .   |   |    !      .   |   |
!     .   |   |    !"
@ 11, 0 SAY "+-------------------+-------------------+-------------------+-----
--------------+"
@ 12, 0 SAY "!     .   |   |    !      .   |   |    !      .   |   |
!      .   |   |    !"
@ 13, 0 SAY "+------------------+-------------------+-------------------+-----
--------------+"
@ 14, 0 SAY "!     .   |   |    !      .   |   |    !      .   |   |
!      .   |   |    !"
@ 15, 0 SAY "+------------------+-------------------+-------------------+-----
--------------+"
@ 18, 0 SAY "Enter as many entries as you want; NOTE that only the row located
by "
@ 19, 0 SAY "the blinking curser can be entered or changed."
@ 20,0 SAY "To delete a entry, leave the  EQUIP NO. of that entry to be '0' ."
@ 21,0 SAY  "NOTE: USE CTRL-W TO SCROLL THROUGH OLD ENTRIES"
FIND &msno&mweek
*      * loop to let user change any data previously entered for this week
DO WHILE (.NOT.EOF).AND.sno=msno.AND.week=mweek.AND.(#<>0)
   @ 4,60 SAY "page " GET pageno PICTURE "9"
   CLEAR GETS
   @ row,2 GET eno1
   @ row,7 GET atchno1
   @ row,12 GET type1
   @ row,15 GET hr1
   @ row,22 GET eno2
   @ row,27 GET atchno2
   @ row,32 GET type2
   @ row,35 GET hr2
   @ row,42 GET eno3
   @ row,47 GET atchno3
   @ row,52 GET type3
   @ row,55 GET hr3
   @ row,62 GET eno4
   @ row,67 GET atchno4
   @ row,72 GET type4
   @ row,75 GET hr4
   READ
   CLEAR GETS
   STORE "1" TO no
   *    * routine to verify that equipment is in eqcost.dbf
   DO WHILE no<= "4"
      IF eno&no=0
         REPLACE atchno&no WITH 0,type&no WITH " " ,hr&no WITH 0
         STORE str(&no+1,1,0) TO no
         LOOP
      ENDIF eno&no=0

      STORE STR(p.eno&no,4,0)+STR(p.atchno&no,2,0) TO meq
      SELECT SECONDARY
      FIND '&meq'


      DO WHILE #=0.AND.eno&no<>0
         SELECT PRIMARY
         @ 16,0 SAY " "+CHR(7)
         ? "        Unknown EQUIPMENT CODE,    try again.   "
         @ row,(&no-1)*20+2 GET eno&no
         @ row,(&no-1)*20+7 GET atchno&no
         @ row,(&no-1)*20+12 GET type&no
         @ row,(&no-1)*20+15 GET hr&no
         READ
         CLEAR GETS
      ENDDO WHILE #=0.AND.eno&no<>0

      SELECT PRIMARY
      IF eno&no=0
         REPLACE atchno&no WITH 0,type&no WITH " " ,hr&no WITH 0
         STORE str(&no+1,1,0) TO no
         LOOP
      ENDIF eno&no=0
      *       * check for illegal type
      DO WHILE !(type&no)<> 'U'.AND.!(type&no)<> 'O' .AND.!(type&no)<>'C'
         @ 16,0 SAY " "
         ? '     Illegal type, must be "U","C" or "O" '
         @ row,(&no-1)*20+12 GET type&no
         READ
         CLEAR GETS
      ENDDO WHILE
      @ 17,0
      STORE str(&no+1,1,0) TO no
   ENDDO WHILE no<= "4"

   STORE row+2 TO row
   IF row>14
      STORE 8 TO row
      STORE pageno+1 TO pageno
   ENDIF row>14

   SKIP 1
ENDDO WHILE (.NOT.EOF).AND.sno=msno.AND.week=mweek.AND.(#<>0)

STORE "Y" TO amore
@ 16,0 SAY "want to add NEW entries of equipment utilized? (y/n)" GET amore
READ
CLEAR GETS
@ 16,0
DO WHILE !(amore)= 'Y'
   SELECT PRIMARY
   APPEND BLANK
   REPLACE sno WITH msno,week WITH mweek,type1 WITH "U"
   REPLACE type2 WITH "U" ,type3 WITH "U" ,type4 WITH "U"
   @ 4,60 SAY "page " GET pageno PICTURE "9"
   CLEAR GETS
   @ row,2 GET eno1
   @ row,7 GET atchno1
   @ row,12 GET type1
   @ row,15 GET hr1
@ row,22 GET eno2
@ row,27 GET atchno2
@ row,32 GET type2
@ row,35 GET hr2
@ row,42 GET eno3
@ row,47 GET atchno3
@ row,52 GET type3
@ row,55 GET hr3
@ row,62 GET eno4
@ row,67 GET atchno4
@ row,72 GET type4
@ row,75 GET hr4
READ
CLEAR GETS
STORE "1" TO no
*       * routine to verify that equipment is in eqcost.dbf
DO WHILE no<= "4"
   IF eno&no=0
      REPLACE atchno&no WITH 0,type&no WITH " " ,hr&no WITH 0
      STORE str(&no+1,1,0) TO no
      LOOP
   ENDIF eno&no=0

   STORE STR(p.eno&no,4,0)+STR(p.atchno&no,2,0) TO meq
   SELECT SECONDARY
   FIND '&meq'


   DO WHILE #=0.AND.eno&no<>0
      SELECT PRIMARY
      @ 16,0 SAY " "+CHR(7)
      ? "        Unknown EQUIPMENT CODE,    try again.   "
      @ row,(&no-1)*20+2 GET eno&no
      @ row,(&no-1)*20+7 GET atchno&no
      @ row,(&no-1)*20+12 GET type&no
      @ row,(&no-1)*20+15 GET hr&no
      READ
      CLEAR GETS
   ENDDO WHILE #=0.AND.eno&no<>0

   SELECT PRIMARY
   IF eno&no=0
      REPLACE atchno&no WITH 0,type&no WITH " " ,hr&no WITH 0
      STORE str(&no+1,1,0) TO no
      LOOP
   ENDIF eno&no=0

   DO WHILE !(type&no)<> 'U' .AND.!(type&no)<> 'O'.AND.!(type&no)<> 'C'
      @ 16,0 SAY " "
      ? '     Illegal type, must be  "U", "C" or "O" '
      @ row,(&no-1)*20+12 GET type&no
      READ
```

```
         CLEAR GETS
      ENDDO WHILE
      @ 17,0
      STORE str(&no+1,1,0) TO no
   ENDDO WHILE no<= "4"


   STORE row+2 TO row
   @ 16,10 SAY "continue? (y/n)" GET amore PICTURE '!'
   READ
   CLEAR GETS

   IF row>14
      STORE 8 TO row
      STORE pageno+1 TO pageno
   ENDIF row>14

ENDDO WHILE !(amore)= 'Y'
*--------------------------------SECTION IV-----------------------------------
-
*       * get other costs
ERASE
@ 2, 18 SAY "SECTION IV: OTHER COST(S) THIS PERIOD"
@ 4, 0 SAY "SCHEDULE NO.: " GET msno
@ 4,26 SAY "TITLE: " GET mtitle
@ 4,54 SAY "PERIOD END :" GET mweek
CLEAR GETS
SELECT PRIMARY
USE wpr INDEX wpr
FIND '&msno&mweek'
@  7, 0 SAY "(18) CSC Labor     : " GET csglbr
@  7,37 SAY "(21) Rental Fees   : " GET rental
@  9, 0 SAY "(19) CSC TDY       : " GET csgtdy
@  9,37 SAY "(22) Transportation: " GET trans
@ 11, 0 SAY "(20) CSC Equipment : " GET csgeq
@ 11,37 SAY "(23) Other US TDY  : " GET other
READ
CLEAR GETS
ERASE
*--------------------------------SECTION V------------------------------------
-
*       * get report of manhours worked on subtasks
@ 2,18 SAY "SECTION V: CONSTRUCTIVE WORK BY SUBTASK"
@ 4, 0 SAY "SCHEDULE NO.: " GET msno
@ 4,26 SAY "TITLE: " GET mtitle
@ 4,54 SAY "PERIOD END :" GET mweek
@  8, 1 SAY "+--------+--------+-----------------+------------+---------+"
@  9, 1 SAY "¢       (25)      ¢       (26)      ¢    (28)    ¢   (31)  ¢"
@ 10, 1 SAY "¢   I    ¢    J   ¢     SUBTASKS    ¢ ACTUAL     ¢ PHY   % ¢"
@ 11, 1 SAY "¢        ¢        ¢                 ¢ MAN HOURS  ¢ COMPLETE ¢"
@ 12, 1 SAY "+--------+--------+-----------------+------------+---------+"
STORE 'wait..one' TO note
```

```
@ 21, 5 GET note
CLEAR GETS
SELECT SECOND
USE act INDEX act
SELECT PRIMARY
USE wact INDEX wact
FIND '&msno&mweek'
@ 21,0
*       * loop to let user change previously entered data
DO WHILE (.NOT.EOF).AND.(p.sno=msno).AND.(mweek=week).AND.(#<>0)
STORE "NOTE: To DELETE and unwanted I-J entry set Hours and Per.Comp. to 0";
TO mnote
@ 7,1 GET mnote
CLEAR GETS
    @ 13, 1 SAY "|        |        |                 |            |
|"
    @ 14, 1 SAY "+--------+--------+-----------------+------------+----------
+"
    @ 15, 1 SAY "|        |        |                 |            |
|"
    @ 16, 1 SAY "+--------+--------+-----------------+------------+----------
+"
    @ 17, 1 SAY "|        |        |                 |            |
|"
    @ 18, 1 SAY "+--------+--------+-----------------+------------+----------
+"
    @ 19, 1 SAY "|        |        |                 |            |
|"
    @ 20, 1 SAY "+--------+--------+-----------------+------------+----------
+"
    @ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field  ;"
    @ 23, 0 SAY "     CTR-W: next screen ;  CTR-D: next character  ;  CTR-S:
next character  ;"
    STORE 1 TO nu
    STORE 13 TO row
    DO WHILE
(nu<5).AND.(.NOT.EOF).AND.(p.sno=msno).AND.(mweek=week).AND.(#<>0)
       @ row, 3 GET p.i
       @ row,12 GET p.j
        STORE msno+STR(p.i,4,0)+STR(p.j,4,0) TO mkey
        SELECT SECONDARY
        FIND '&mkey'
          @ row,21 GET s.des
        CLEAR GETS
        SELECT PRIMARY
          @ row,39 GET p.amh
          @ row,52 GET p.per
          READ
          CLEAR GETS
          STORE nu+1 TO nu
          STORE row+2 TO row
          SKIP +1
    ENDDO while (nu<5)
ENDDO
@ 7,0
STORE 'NOTE: Enter 0 for Start Node to STOP inputting' to note
@  7,10 GET note
CLEAR GETS
STORE "Y" TO mmore
DO WHILE !(mmore) <> "N"
      @ 21, 5
      STORE "Y" TO mmore
      STORE "want to add new i-j's ?(y/n)" TO note
      @ 21, 5
      @ 21, 5 GET note
      CLEAR GETS
      @ 21,34 GET mmore
      READ
      @ 21, 5
      CLEAR GETS
      IF !(mmore)<>"Y"
         STORE "N" TO mmore
         LOOP
      ENDIF
    @ 13, 1 SAY "|        |        |                 |            |
|"
    @ 14, 1 SAY "+--------+--------+-----------------+------------+----------
+"
    @ 15, 1 SAY "|        |        |                 |            |
|"
    @ 16, 1 SAY "+--------+--------+-----------------+------------+----------
+"
    @ 17, 1 SAY "|        |        |                 |            |
|"
    @ 18, 1 SAY "+--------+--------+-----------------+------------+----------
+"
    @ 19, 1 SAY "|        |        |                 |            |
|"
    @ 20, 1 SAY "+--------+--------+-----------------+------------+----------
+"
    @ 22, 0 SAY "Type CTR-F: next field  ;  CTR-A: previous field  ;"
    @ 23, 0 SAY "     CTR-W: next screen ;  CTR-D: next character  ;  CTR-S:
next character  ;"
    STORE 1 TO nu
    STORE 13 TO row
    DO WHILE (nu<5)
       STORE 0 TO mi
       @ row, 3 GET mi PICTURE '9999'
       READ
        @ 21,0
       CLEAR GETS
       IF mi=0
          STORE 5 TO nu
          STORE 'N' TO mmore
          LOOP
       ENDIF
       STORE 0 TO mj
       @ row,12 GET mj PICTURE '9999'
       READ
       CLEAR GETS
       STORE STR(mi,4,0)+STR(mj,4,0) TO mij
       STORE msno+mweek+mij TO mwkey
       STORE msno+mij TO mkey
       SELECT SECONDARY
       FIND "&mkey"
       IF #=0
          *      * task not in data base
          STORE 'I-J NOT FOUND'+CHR(7) TO note
          @ 21,5 GET note
          CLEAR GETS
          LOOP
       ENDIF
       SELECT PRIMARY
       FIND '&mwkey'
        IF #=0
          APPEND BLANK
          REPLACE i WITH mi, j WITH mj, sno WITH msno, week WITH mweek
        ENDIF
          @ row,21 GET des
          CLEAR GETS
          @ row,39 GET p.amh
          @ row,52 GET p.per
          READ
          CLEAR GETS
          STORE nu+1 TO nu
          STORE row+2 TO row
    ENDDO while (nu<5)

 ENDDO while !(mmore)<>"N"

ERASE
CLEAR GETS
RETURN
*
*       * END WPR2.CMD
*
```

APPENDIX B:

DATA BASE STRUCTURES

CONTENTS

ACT.DBF

STRUCTURE FOR FILE: ACT.DBF

Data base to store the Critical Path Method activities that make up the project. The project officer records these activities on the Weekly Progress Report form. The operators input the activity data using the IWPR.CMD program. This is a master file updated by the transaction file WACT.DBF during processing by PS.CMD

```
FLD        NAME       TYPE WIDTH   DEC    COMMENTS
001       SNO          C     006          Schedule Number
002       I            N     004          Start Node
003       J            N     004          End Node
004       DES          C     015          Description
005       SMH          N     005          Scheduled Manhours to Complete
006       AMH          N     005          Actual Manhours
007       PER          N     003          Percent Complete
** TOTAL **                 00043
```

INDEX FILE: ACT.NDX

KEY EXPRESSION: sno+STR(I,4,0)+STR(J,4,0)

STRUCTURE FOR FILE: BILL.DBF

This data base stores the cumulatives for the customer billing report generated by BILLRPT. The user establishes the billing data base through the BILL.CMD program. The input to this program is the Project Number (pno) from the Construction Directive and other miscellaneous data such as the title, unit, DA 2544 number, etc. Each project in the Project Data Base with the same pno will be used to produce the billing report. The file is updated when the report is generated.

FLD	NAME	TYPE	WIDTH	DEC	COMMMENTS
001	PNO	C	014		
002	BILLNO	N	002		Bill Number
003	FROM	C	006		Starting Date for Report
004	TO	C	006		End Date for Report
005	DA	C	010		DA 2544 Number
006	DATE	C	006		Date of DA 2544
007	UNIT	C	005		Battalion Identifier
008	TITLE	C	015		Project Title
009	USTDY	N	010	002	US TDY Costs
010	USEQUIP	N	010	002	US Equipment Costs
011	CSGTDY	N	010	002	Civilian Service Group TDY Costs
012	CSGEQUIP	N	010	002	Civilian Service Group Equip. Costs
013	CSGLBR	N	010	002	Civilian Service Group Labor Costs
** TOTAL **			00115		

INDEX FILE: BILL.NDX

KEY EXPRESSION: pno

STRUCTURE FOR FILE: BILL.DBF

This data base stores the cumulatives for the customer billing report generated by BILLRPT. The user establishes the billing data base through the BILL.CMD program. The input to this program is the Project Number (pno) from the Construction Directive and other miscellaneous data such as the title, unit, DA 2544 number, etc. Each project in the Project Data Base with the same pno will be used to produce the billing report. The file is updated when the report is generated.

FLD	NAME	TYPE	WIDTH	DEC	COMMMENTS
001	PNO	C	014		
002	BILLNO	N	002		Bill Number
003	FROM	C	006		Starting Date for Report
004	TO	C	006		End Date for Report
005	DA	C	010		DA 2544 Number
006	DATE	C	006		Date of DA 2544
007	UNIT	C	005		Battalion Identifier
008	TITLE	C	015		Project Title
009	USTDY	N	010	002	US TDY Costs
010	USEQUIP	N	010	002	US Equipment Costs
011	CSGTDY	N	010	002	Civilian Service Group TDY Costs
012	CSGEQUIP	N	010	002	Civilian Service Group Equip. Costs
013	CSGLBR	N	010	002	Civilian Service Group Labor Costs
** TOTAL **			00115		

INDEX FILE: BILL.NDX

KEY EXPRESSION: pno

STRUCTURE FOR FILE: BN.DBF

Data base used to produce the Battalion Cost Report by the program BNRPT.CMD. One record is created in the Project Data Base (PDFMNU.CMD) for each battalion. The file is updated when the report is generated. Only one record is maintained for each battalion.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	BILLNO	N	002		Bill Number
002	FROM	C	006		Starting Date for Report
003	TO	C	006		End Date for Report
004	BN	C	004		Battalion Identifier
005	USTDY	N	010	002	US TDY Costs
006	USEQUIP	N	010	002	US Equipment Costs
007	CSGTDY	N	010	002	Civilian Service Group TDY Costs
008	CSGEQUIP	N	010	002	Civilian Service Group Equipment Costs
009	CSGLBR	N	010	002	Civilian Service Group Labor Costs
** TOTAL **			00069		

INDEX FILE: BN.NDX

KEY EXPRESSION: bn

STRUCTURE FOR FILE: CLUSTERS.DBF

Data structure used in CLUSTERS.CMD to implement the project reporting clusters. A cluster is a set of groups. A group is a set of schedule numbers. The clusters data base is the root (cno) of a tree whose level-one children are groups. The group's children are schedule numbers. A group may have up to 99 children.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	CNO	N	001		Cluster Number
002	CNAME	C	020		Cluster Name
** TOTAL **			00022		

INDEX FILE: CLUSTERS.NDX

KEY EXPRESSION: STR(cno,1,0)

STRUCTURE FOR FILE: DAYS.DBF

This data base contains the six digit dates of the form MMDDYY for the period end of the Weekly Progress Reports. If the period end is Friday, then this file would contain all Fridays for that year. It may contain more than one year.

```
FLD        NAME       TYPE WIDTH   DEC   COMMENTS
001      DAY           C     006         MMDDYY:Period End
** TOTAL **                00007
```

INDEX FILE: DAYS.NDX

KEY EXPRESSION: day

STRUCTURE FOR FILE: EQ.DBF

The structure of this file is the same as that of WEQ.DBF WEQA.DBF, and WEQB.DBF. This file is used to store all the old weekly equipment usage records reported on the Weekly Progress Report. It is updated by the BACKUP.CMD program by appending from WEQ.DBF. The file is used by the program EQAUDIT.CMD to print all past equipment usage of a project for auditing.

```
FLD       NAME      TYPE WIDTH   DEC    COMMMENTS
001     SNO          C    006           Project Schedule Number
002     WEEK         C    006           WPR Reporting Period
003     ENO1         N    004           Equipment Item Number
004     ATCHNO1      N    002           Attachment Number
005     TYPE1        C    001           US, Civilian Service, or Other
006     HR1          N    004           Number of Hours Used
007     ENO2         N    004
008     ATCHNO2      N    002
009     TYPE2        C    001
010     HR2          N    004
011     ENO3         N    004
012     ATCHNO3      N    002
013     TYPE3        C    001
014     HR3          N    004
015     ENO4         N    004
016     ATCHNO4      N    002
017     TYPE4        C    001
018     HR4          N    004
** TOTAL **             00057
```

INDEX FILE: EQ.NDX

KEY EXPRESSION: sno+week

STRUCTURE FOR FILE: EQCOST.DBF

The equipment data base is contained in this file. It is created and maintained by the EQCOST.CMD program and is used to calculate the equipment costs for each project. Based on the Weekly Progress Report input of item number (eno and/or atchno) and the number of hours used, the main processing program PS.CMD calculates the weekly equipment cost for the project by finding the OMCOST in EQCOST.DBF. Equipment may have attachments, i.e, subcomponents. The acthno field is used for this purpose.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	ENO	N	004		Equipment Item Number
002	ATCHNO	N	002		Equipment Attachment Number
003	DESP1	C	030		Description Field 1
004	ACOST	N	010	002	Acquisition Cost
005	DCOST	N	005	002	Depreciation Cost
006	OMCOST	N	005	002	Operation and Maintenance Cost
007	DESP2	C	030		Description Field 2
008	DESP3	C	015		Description Field 3
009	TYPE	C	001		US/Local/Special
010	F	C	001		On AMDF:boolean
011	R	C	001		Costs Reportable:boolean
012	A	C	004		Federal Stock Number-A
013	B	C	002		FSN-B
014	C	C	003		FSN-C
015	D	C	004		FSN-D
016	N	C	001		Attachment Type:A=Attachment I=Individual Component N=None
** TOTAL **			00119		

INDEX FILE: EQCOST.NDX

KEY EXPRESSION: STR(eno,4,0)+STR(atchno,2,0)

STRUCTURE FOR FILE: GROUPS.DBF

Provides the data structure for groups definition in the GROUPS.CMD program. Each group has one record that contains the cluster number to which it belongs.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	CNO	N	001		Cluster Number
002	GNO	N	002		Group Number
003	GNAME	C	020		Group Name
004	GMAN	C	010		Group Manager
** TOTAL **			00034		

INDEX FILE: GROUPS.NDX

KEY EXPRESSION: STR(cno,1,0)+STR(gno,2,0)

STRUCTURE FOR FILE: LOCKSA.DBF

Implements a crude record lock for two-user (A and B) input to Weekly Progress Report. WPR.CMD stores user's current schedule number and week in this file and checks the other user's lock file to insure they are not inputting for the same project. There is a LOCKSB.DBF for the B terminal user.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	WSLCK	L	001		NOT USED
002	SNOWEEK	C	012		String = schedule+week
003	WPRPSLCK	L	001		NOT USED
** TOTAL **			00015		

STRUCTURE FOR FILE: MANDAYS.DBF

Master file to store mandays data. One record per project is maintained and updated during processing by PS.CMD from the transaction file WMANDAYS.DBF.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	SNO	C	006		Project Schedule Number
002	OSUPPORT	N	006		Officer Support TDY Mandays
003	NSUPPORT	N	006		NCO Support TDY Mandays
004	ESUPPORT	N	006		Enlisted Support TDY Mandays
005	OCHARGE	N	006		Officer TDY Mandays Charged To Project
006	NCHARGE	N	006		NCO Support TDY Charged To Project
007	ECHARGE	N	006		Enlisted Support TDY Charged To Project
008	OWORK	N	006		Officer Mandays Worked On Site
009	NWORK	N	006		NCO Mandays Worked On Site
010	EWORK	N	006		Enlisted Mandays Worked On Site
011	WLOST	N	006		Mandays Lost Due To Weather
012	MLOST	N	006		Mandays Lost Due To Maitenance
013	OLOST	N	006		Mandays Lost Due To Other Reasons
** TOTAL **			00079		

INDEX FILE: MANDAYS.NDX

KEY EXPRESSION: sno

STRUCTURE FOR FILE: MISC.DBF

This data base consists of one record of system control information.

```
FLD        NAME       TYPE WIDTH   DEC   COMMENTS
001       BN            N    004         NOT USED
002       GOFF          N    005   002   Default Officer TDY Rate
003       GNCO          N    005   002   Default NCO TDY Rate
004       GEM           N    005   002   Default Enlisted TDY Rate
005       LASTMONTH     C    006         Not Used
006       DATE          C    006         Date of Last WPR Processing
007       MISC1         N    010         NOT USED
008       MISC2         N    010         NOT USED
009       MISC3         C    020         NOT USED
** TOTAL **               00072
```

STRUCTURE FOR FILE: PDF.DBF

This data base stores the Project Data Form data except for Allocated Funds, which is stored in PS.DBF. One record is created for each project by PDFMNU.CMD. This is the first instance of the project in the system and a record for each project must be in this data base before any input or processing can take place.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	PNO	C	014		Project No. from The Const. Directive
002	SNO	C	006		Project Schedule Number
003	TITLE	C	020		Title of the Project
004	LOC	C	015		Location
005	BN	C	004		Battalion
006	CO	C	001		Company
007	PLT	N	001		Platoon
008	PJTYPE	C	001		Project Funding Type:MCA,MMCA,OMA, ect.
009	ECOST	N	010	002	Estimatated Contractor Cost
010	ETCOST	N	010	002	Estimated Troop Construction Cost
011	SDATE	C	006		Construction Start Date
012	CDATE	C	006		Construction Completion Date
013	CUSTOMER	C	020		Customer
014	COMMENTS	C	069		Any comments the user might have
015	GRAF	C	001		Use Default TDY Rates:boolean
016	OFF	N	005	002	Non-Default Officer TDY Rates
017	NCO	N	005	002	Non-Default NCO TDY Rates
018	EM	N	005	002	Non-Default Enlisted TDY Rates
019	ESC1	N	010	002	Estimated Cost for US TDY
020	ESC2	N	010	002	Estimated Cost for US Equipment
021	ESC3	N	010	002	Estimated Cost for CSG TDY
022	ESC4	N	010	002	Estimated Cost for CSG Equipment
023	ESC5	N	010	002	Estimated Cost for CSG Labor
** TOTAL **			00250		

INDEX FILE: PDF.NDX

KEY EXPRESSION: sno

STRUCTURE FOR FILE: PS.DBF

This is the Project Status master file updated during the main processing program PS.CMD. Also contains three fields used to produce the Monthly Contractor and Troop Construction Cost Report (MCTRPT.CMD): pper, pfunds, and pdirdays. These fields store the values accumulated since the last MCTRPT.CMD.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	SNO	C	006		Schedule Number
002	DATE	C	006		Date of Last Update
003	BN	C	004		Battalion
004	CO	C	001		Company
005	PLT	N	001		Platoon
006	ADJUST	N	004		+ or - Days on the Critical Path
007	CWS	N	006	002	Scheduled Percent Complete
008	SMH	N	008		Total Scheduled Manhours
009	AMH	N	008		Total Actual Manhours
010	ACTPER	N	006	002	Actual Percent Complete
011	AC1	N	010	002	Allocated Funds for US TDY
012	AC2	N	010	002	Allocated Funds for US Equipment
013	AC3	N	010	002	Allocated Funds for CSG TDY
014	AC4	N	010	002	Allocated Funds for CSG Equipment
015	AC5	N	010	002	Allocated Funds for CSG Labor
016	EC1	N	010	002	Expended to Date for US TDY
017	EC2	N	010	002	Expended to Date for US Equipment
018	EC3	N	010	002	Expended to Date for CSG TDY
019	EC4	N	010	002	Expended to Date for CSG Equipment
020	EC5	N	010	002	Expended to Date for CSG Labor
021	PPER	N	006	002	Percent Complete Since Last MCTRPT.CMD
022	PFUNDS	N	011	002	Total Funds Expended Since MCTRPT.CMD
023	PDIRDAYS	N	006		Direct Const. Mandays Since MCTRPT.CMD
** TOTAL **			00174		

INDEX FILE: PS.NDX

KEY EXPRESSION: sno

STRUCTURE FOR FILE: SCHEDULE.DBF

Data structure used in SCHEDULE.CMD to implement project reporting groups. The groups are sets of schedule numbers over which reports will be generated and rolled up to produce summaries. A group is a unique, mutually exclusive set of schedule numbers within a cluster, and a cluster is a set of groups. The data base contains links to its parent group and cluster.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	CNO	N	001		Cluster Number
002	GNO	N	002		Group Number
003	XNO	N	002		Number of Schedule Within the Group
004	SNO	C	006		Schedule Number
** TOTAL **			00012		

INDEX FILE: SCHEDULE.NDX

KEY EXPRESSION: STR(cno,1,0)+STR(gno,2,0)+STR(xno,2,0)

WACT.DBF

STRUCTURE FOR FILE: WACT.DBF

This is a "master transaction" file for the transaction files WACTA.DBF and WACTB.DBF which store the Weekly Progress Report data for manhours of work on Critical Path Method i-j activities. WACT.DBF is produced by appending WACTA.DBF and WACTB.DBF and is used by the processing program PS.CMD to calculate the actual percent complete and to update the master file ACT.DBF.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	SNO	C	006		Project Schedule Number
002	WEEK	C	006		Weekly Progress Report Period
003	I	N	004		Start Node
004	J	N	004		End Node
005	AMH	N	005		Actual Manhours Worked
006	PER	N	003		Actual Percent Complete
** TOTAL **			00029		

INDEX FILE: WACT.NDX

KEY EXPRESSION: sno+week+STR(i,4,0)+STR(j,4,0)

STRUCTURE FOR FILE: WEQ.DBF

The "master transaction" file for Weekly Progress Report input of equipment usage by project. This file is produced by appending from WEQA.DBF and WEQB.DBF in the main processing program PS.CMD. Note the structure consists of four equipment "subrecords."

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	SNO	C	006		Project Schedule Number
002	WEEK	C	006		WPR Reporting Period
003	ENO1	N	004		Equipment Item Number
004	ATCHNO1	N	002		Attachment Number
005	TYPE1	C	001		US, Civilian Service, or Other
006	HR1	N	004		Number of Hours Used
007	ENO2	N	004		
008	ATCHNO2	N	002		
009	TYPE2	C	001		
010	HR2	N	004		
011	ENO3	N	004		
012	ATCHNO3	N	002		
013	TYPE3	C	001		
014	HR3	N	004		
015	ENO4	N	004		
016	ATCHNO4	N	002		
017	TYPE4	C	001		
018	HR4	N	004		
** TOTAL **			00057		

INDEX FILE: WEQ.NDX

KEY EXPRESSION: sno+week

STRUCTURE FOR FILE: WMANDAYS.DBF

"Master transaction" file updated during initial phase of PS.CMD by appending from the transaction files WMDAYSA.DBF and WMDAYSB.DBF. This file then becomes the transaction file for updating the master MANDAYS.DBF.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	SNO	C	006		Project Schedule Number
002	WEEK	C	006		Officer Support TDY Mandays
003	OSUPPORT	N	006		NCO Support TDY Mandays
004	NSUPPORT	N	006		Enlisted Support TDY Mandays
005	ESUPPORT	N	006		Officer TDY Mandays Charged To Project
006	OCHARGE	N	006		NCO Support TDY Charged To Project
007	NCHARGE	N	006		Enlisted Support TDY Charged To Project
008	ECHARGE	N	006		Officer Mandays Worked On Site
009	OWORK	N	006		NCO Mandays Worked On Site
010	NWORK	N	006		Enlisted Mandays Worked On Site
011	EWORK	N	006		Mandays Lost Due To Weather
012	WLOST	N	006		Mandays Lost Due To Maintenance
013	MLOST	N	006		Mandays Lost Due To Other Reasons
014	OLOST	N	006		
** TOTAL **			00085		

INDEX FILE: WMANDAYS.NDX

KEY EXPRESSION: sno+week

STRUCTURE FOR FILE: WPR.DBF

The master transation file for miscellaneous data input in WPR.CMD. Created in PS.CMD by appending from transaction files WPRA.DBF and WPRB.DBF.

FLD	NAME	TYPE	WIDTH	DEC	COMMENTS
001	SNO	C	006		Schedule Number
002	WEEK	C	006		Weekly Report Period
003	ADJUST	N	003		+ or - days on the Critical Path
004	CWS	N	005	002	Schedule Percent Complete
005	CSGLBR	N	008	002	CSG Labor in Dollars
006	CSGTDY	N	008	002	CSG TDY in Dollars
007	CSGEQ	N	008	002	CSG Equipment in Dollars
008	RENTAL	N	008	002	Rental Costs
009	TRANS	N	008	002	Transportaion Costs
010	OTHER	N	008	002	Other Costs (accrues to US TDY)
** TOTAL **			00069		

INDEX FILE: WPR.NDX

KEY EXPRESSION: sno+week

APPENDIX C:

DATA BASE USAGE IN COMMAND FILES

CONTENTS

```
LEVEL   COMMAND DATABASE AND INDEX

 0      backup
                USE misc
                USE wpr INDEX wpr
                USE tt
                USE wpr
                USE wmandays INDEX wmandays
                USE tt
                USE wmandays
                USE wact INDEX wact
                USE tt
                USE wact
                USE weq INDEX weq
                USE eq INDEX eq
                USE tt
                USE weq
                USE wpr&mmdd
                USE wman&mmdd
                USE weq&mmdd
                USE wact&mmdd
```

COMMENTS:

tt is temporary database
mmdd is memory variable; month and day.

BILL.CMD

```
LEVEL   COMMAND DATABASE AND INDEX

 0      bill
                USE bill INDEX bill
                USE pdf
```

BILLRPT.CMD

```
LEVEL   COMMAND DATABASE AND INDEX

 0      billrpt
                USE misc
                USE pdf INDEX pdf
                USE bill INDEX bill
                USE ps INDEX ps
                USE bill INDEX bill
```

BNRPT.CMD

```
LEVEL   COMMAND DATABASE AND INDEX


 0      bnrpt
                USE misc
                USE pdf
                USE ps INDEX ps
                USE bn INDEX bn
                USE pdf INDEX pdf.bn
```

CLUSTERS.CMD

```
LEVEL   COMMAND DATABASE AND INDEX

 0      clusters
                USE clusters INDEX clusters
 1              groups
                    USE groups INDEX groups
 2                  schedule
                       USE pdf INDEX pdf
                       USE schedules INDEX schedules
                       USE groups INDEX groups
                       USE schedule INDEX schedule
                       USE groups INDEX groups
                       USE schedule INDEX schedule
                       USE schedule INDEX schedule
                       USE pdf INDEX pdf
                       USE groups INDEX groups
0
               USE clusters
               USE b:clusters
               USE groups
               USE b:groups
               USE schedule
               USE b:schedule
```

COSTRPT.CMD

```
LEVEL   COMMAND DATABASE AND INDEX

 0      costrpt
                USE misc
                USE clusters INDEX clusters
                USE groups INDEX groups
                USE schedule INDEX schedule
                USE tsno
                USE groups INDEX groups
                USE ps INDEX ps
```

COMMENTS:

tsno is a temporary file containing schedule numbers included in report.

```
LEVEL   COMMAND DATABASE AND INDEX

 0      eqaudit
                USE eq INDEX eq
                USE pdf INDEX pdf
                USE eqcost INDEX eqcost
```

EQCOST.CMD

```
LEVEL   COMMAND DATABASE AND INDEX

 0      eqcost
                USE eqcost INDEX eqcost
 1              edeq
 2                 edeqa
 1              edeqa
 1              edeq
 2                 edeqa
 1              edeq
 2                 edeqa
 1              edeqa
 1              eqrpt
                   USE
                   USE eqcost
                   USE teq
                   USE
                USE eqcost
                USE b:eqcost
```

```
LEVEL   COMMAND DATABASE AND INDEX

 0      iwpr
                USE act INDEX act
                USE pdf INDEX pdf
                USE ps INDEX ps
                USE pdf INDEX pdf
                USE ps INDEX ps
                USE pdf INDEX pdf
                USE ps INDEX ps
                USE pdf INDEX pdf
                USE b:act
```

MCTRPT.CMD

```
LEVEL   COMMAND DATABASE AND INDEX


 0      mctrpt
                USE misc
                USE pdf
                USE mandays INDEX mandays
                USE ps INDEX ps
```

```
LEVEL   COMMAND DATABASE AND INDEX


 0      pdfmnu
            USE pdf INDEX pdf
            USE ps INDEX ps
            USE misc
 1          pdf
 2              pdf1
 3                  pdf2
                        USE ps INDEX ps
                        USE mandays INDEX mandays
                        USE ps INDEX ps
                        USE bn INDEX bn
                        USE ps INDEX ps
 2              pdf1
 3                  pdf2
                        USE ps INDEX ps
                        USE mandays INDEX mandays
                        USE ps INDEX ps
                        USE bn INDEX bn
                        USE ps INDEX ps
            USE ps INDEX ps
            USE schedule INDEX schedule
            USE groups INDEX groups
 1          pdfmnu2
                USE pdf INDEX pdf
 2              listpdf
                    USE pdf INDEX pdf
                    USE tpdf
 3                  ppdf
                        USE ps INDEX ps
                    USE tpdf
 3                  sppdf
                        USE ps INDEX ps
                        USE pdf INDEX pdf
                    USE tpdf
                    USE ps INDEX ps
 2              listpdf
                    USE pdf INDEX pdf
                    USE tpdf
 3                  ppdf
                        USE ps INDEX ps
                        USE tpdf
 3                  sppdf
                        USE ps INDEX ps
                        USE pdf INDEX pdf
                        USE tpdf
                        USE ps INDEX ps
 2              listpdf
                    USE pdf INDEX pdf
                    USE tpdf
```

```
3                   ppdf
                        USE ps INDEX ps
                        USE tpdf
3                   sppdf
                        USE ps INDEX ps
                        USE pdf INDEX pdf
                        USE tpdf
                        USE ps INDEX ps
                        USE pdf INDEX pdf
2               ppdf
                    USE ps INDEX ps
2               sppdf
                    USE ps INDEX ps
                    USE pdf INDEX pdf
```

PS.CMD

```
LEVEL   COMMAND DATABASE AND INDEX

 0      ps
                USE wpr INDEX wpr
                USE wmandays INDEX wmandays
                USE wact INDEX wact
                USE a:weq INDEX weq
                USE wpra
                USE twpra
                USE twprb
                USE wmdaysa
                USE twmdaysa
                USE twmdaysb
                USE wacta
                USE twacta
                USE twactb
                USE weqa
                USE tweqa
                USE tweqb
                USE misc
                USE days INDEX days
                USE misc
                USE ps INDEX ps
                USE wpr INDEX wpr
                USE pdf INDEX pdf
                USE wmandays INDEX wmandays
                USE wmandays INDEX wmandays
                USE eqcost INDEX eqcost
                USE weq INDEX weq
                USE wact INDEX wact
                USE act INDEX act
                USE wmandays INDEX wmandays
                USE mandays INDEX mandays
                USE ps INDEX ps
                USE wpr INDEX wpr
```

COMMENTS:

t* files are temporary files used to clear the transaction files w*a and w*b.

PSR.CMD

```
LEVEL   COMMAND DATABASE AND INDEX

 0      psr
                USE misc
                USE clusters INDEX clusters
                USE groups INDEX groups
                USE schedules INDEX schedules
                USE schedules INDEX schedules
                USE schedules INDEX schedules
                USE ps INDEX ps
                USE tsno
                USE ps INDEX ps
                USE tsno
```

COMMENTS:

tsno is a temporary file containing a list of schedule numbers for report generation.

```
LEVEL   COMMAND DATABASE AND INDEX

0       psrpt
                USE pdf INDEX pdf
                USE a:mandays INDEX a:mandays
                USE a:wmandays INDEX a:wmandays
                USE ps INDEX ps
                USE eqcost INDEX eqcost
                USE a:weq INDEX a:weq
```

S3RPT.CMD

```
LEVEL   COMMAND DATABASE AND INDEX

 0      s3rpt
                USE misc
                USE clusters INDEX clusters
                USE groups INDEX groups
                USE schedule INDEX schedule
                USE schedule INDEX schedule
                USE tsno
                USE tsno INDEX tsno
                USE ps INDEX ps
                USE groups INDEX groups
                USE pdf INDEX pdf
                USE ps INDEX ps
```

COMMENTS:

tsno is a temporary file containing a list of schedule numbers for report generation.

WCSRPT.CMD

```
LEVEL   COMMAND DATABASE AND INDEX


 0      wcsrpt
                USE misc
                USE clusters INDEX clusters
                USE groups INDEX groups
                USE schedule INDEX schedule
                USE schedules INDEX schedules
                USE tsno
                USE groups INDEX groups
                USE pdf INDEX pdf
                USE ps INDEX ps
```

COMMENTS:

tsno is a temporary file containing a list of schedule numbers for report generation.

WPR.CMD

```
LEVEL   COMMAND DATABASE AND INDEX


 0      wpr
            USE &my.:ps INDEX &my.:ps
            USE &my.:days INDEX &my.:days
            USE &my.:pdf INDEX &my.:pdf
            USE locks&else
            USE wpr&else INDEX wpr&else
            USE locks&my
            USE wpr&my INDEX wpr&my
            USE wpr&my INDEX wpr&my
            USE wmdays&my INDEX wmdays&my
            USE wpr&my INDEX wpr&my
 1          wpr1
                USE wmdays&my INDEX wmdays&my
 2              wpr2
                    USE &my.:eqcost INDEX &my.:eqcost
                    USE weq&my INDEX weq&my
                    USE wpr&my INDEX wpr&my
                    USE &my.:act INDEX &my.:act
                    USE wact&my INDEX wact&my
            USE locks&my
```

COMMENTS:

else and my are memory variables identifying the user.

CERL DISTRIBUTION

Chlof of Engineers
ATTN: Tech Monitor
ATTN: DAEN-ASI-L (2)
ATTN: DAEN-CCP
ATTN: DAEN-CW
ATTN: DAEN-CWE
ATTN: DAEN-CWM-R
ATTN: DAEN-CWO
ATTN: DAEN-CWP
ATTN: DAEN-EC
ATTN: DAEN-ECC
ATTN: DAEN-ECE
ATTN: DAEN-ZCF
ATTN: DAEN-ECR
ATTN: DAEN-RD
ATTN: DAEN-RDC
ATTN: DAEN-RDM
ATTN: DAEN-RM
ATTN: DAEN-ZCZ
ATTN: DAEN-ZCE
ATTN: DAEN-ZCI
ATTN: DAEN-ZCM

FESA, ATTN: Library 22060
ATTN: DET III 79906

US Army Engineer Districts
ATTN: Library (41)

US Army Engineer Divisions
ATTN: Library (14)

US Army Europe
AEAEN-ODCS/Engr 09403
ISAE 09081
V Corps
ATTN: DEH (11)
VII Corps
ATTN: DEH (15)
21st Support Command
ATTN: DEH (12)
USA Berlin
ATTN: DEH (15)
USASETAF
ATTN: DEH (6)
Allied Command Europe (ACE)
ATTN: DEH (3)

8th USA, Korea (14)

ROK/US Combined Forces Command 96301
ATTN: EUSA-HHC-CFC/Engr

USA Japan (USARJ)
ATTN: AJEN-FE 96343
ATTN: DEH-Honshu 96 '3
ATTN: DEH-Okinawa 96331

Rocky Mt. Area 80903

Area Engineer, AEDC-Area Office
Arnold Air Force Station, TN 37389

Western Area Office, CE
Vandenberg AFB, CA 93437

416th Engineer Command 60623
ATTN: Facilities Engineer

US Military Academy 10966
ATTN: Facilities Engineer
ATTN: Dept of Geography &
Computer Science
ATTN: DSCPER/MAEN-A

AMMRC, ATTN: DRXMR-WE 02172

USA ARRCOM 61299
ATTN: DRCIS-RI-I
ATTN: DRSAR-IS

DARCOM - Dir., Inst., & Svcs.
ATTN: DEH (23)

DLA ATTN: DLA-WI 22314

FORSCOM
FORSCOM Engineer, ATTN: AFEN-DEH
ATTN: DEH (23)

HSC
ATTN: HSLO-F 78234
ATTN: Facilities Engineer
Fitzsimons AMC 80240
Walter Reed AMC 20012

INSCOM - Ch, Instl. Div.
ATTN: Facilities Engineer (3)

MDW
ATTN: DEH (3)

MTMC
ATTN: MTMC-SA 20315
ATTN: Facilities Engineer (3)

NARADCOM, ATTN: DRDNA-F 071160

TARCOM, Fac. Div. 48090

TRADOC
HQ, TRADOC, ATTN: ATEN-DEH
ATTN: DEH (19)

TSARCOM, ATTN: STSAS-F 63120

USACC
ATTN: Facilities Engineer (2)

WESTCOM
ATTN: DEH
Fort Shafter 96858
ATTN: APEN-IM

SHAPE 09055
ATTN: Survivability Section, CCB-OPS
Infrastructure Branch, LANDA

HQ USEUCOM 09128
ATTN: ECJ 4/7-LOE

U.S. Army, Fort Belvoir 22060
ATTN: Canadian Liaison Officer
ATTN: Water Resources Support Center
ATTN: Engr Studies Center
ATTN: Engr Topographic Lab
ATTN: ATZA-DTE-SU
ATTN: ATZA-DTE-EM
ATTN: R & D Command

CRREL, ATTN: Library 03755

ETL, ATTN: Library 22060

WES, ATTN: Library 39180

HQ, XVIII Airborne Corps and
Ft. Bragg 28307
ATTN: AFZA-FE-EE

Chanute AFB, IL 61868
3345 CES/DE, Stop 27

Norton AFB CA 92409
ATTN: AFRCE-MX/DEE

Tyndall AFB, FL 32403
AFESC/Engineering & Service Lab

NAVFAC
ATTN: RDT&E Liaison Office (6)
ATTN: Sr. Tech. FAC-03T 22332
ATTN: Asst. CDR R&D, FAC-03 22332

NCEL 93041
ATTN: Library (Code L08A)

Defense Technical Info. Center 22314
ATTN: DDA (12)

Engineering Societies Library
New York, NY 10017

National Guard Bureau 20310
Installation Division

US Government Printing Office 22304
Receiving Section/Depository Copies (2)

US Army Env. Hygiene Agency
ATTN: HSHB-E 21010

National Bureau of Standards 20760

300
1/9/84

Special Distribution

USMA
ATTN: D/G&CS

USAES
ATTN: ATZA-CD
ATTN: ATZA-TDO-CTT
ATTN: ATZA-DTE
ATTN: ATZA-FM

Commanding General
35th Engineer Brigade (Corps)
Sherman Road
Jefferson Barracks, MO 63125

Commander
30th Engineer Brigade
North Carolina National Guard
ATTN: LTC S. L. Camacho
Charlotte, NC 28200

Commander
CECOM Software Development
Support Center
ATTN: DRCPM-OTDS-SDSC/Mr. Gary Dozier
Building 138
Fort Leavenworth, KS 66027

Commander
7th Engineer Brigade
APO New York 09154

Commander
18th Engineer Brigade
ATTN: MAJ Baldwin
APO New York 09164

Commander
18th Engineer Brigade
APO New York 09164

Commander
20th Engineer Brigade
ATTN: Automation Officer
Fort Bragg, NC 28307

Commander
130th Engineer Brigade
APO New York 09165

Commander
2nd Engineer Group
ATTN: Automation Officer
APO San Francisco 96301

CDR USACE (DAEN-RDM)
WASH DC 20314

Commander
Area Army, COE
Area B Dubling 8, Room 98
Wright Patterson AFB, OH 45433

Commander
36th Engineer Group
Fort Benning, GA 31905

Commander
937th Engineer Group
Fort Riley, KS 66442

Commander
High Technology Test Bed
9th Infantry Division
ATTN: AFYO-TD-FDCS/MAJ Barber
Fort Lewis, WA 98432

CDR USACE (DAEN-ZCM/Dr. Meyer)
WASH DC 20314

Commander
U.S. Army Combined Arms Center
ATTN: ATZL-CAC-IA/CPT Nelson
ATTN: ATZL-CAC-C/CPT Lynn
Fort Leavenworth, KS 66027

Commander
U.S. Army Communications and
Electronics Command
ATTN: DAC-PM-OTDS-SDSC/MAJ Harris
Fort Leavenworth, KS 66027

Commander
U.S. Army Engineer Division, Europe
ATTN: Chief ADP/Mr. Dahlen
APO New York 09757

Commander
USARMR VIII
ATTN: COL John Coats
Fitzsimons Army Medical Center
Aurora, CO 80045

Commander
Naval Construction Battalion Center
ATTN: Code 152/Mr. Beringer
Port Hueneme, CA 93043

Superintendent
U.S. Military Academy
ATTN: MADN-F
West Point, NY 10996

Superintendent
U.S. Air Force Academy
ATTN: DFCE/MAJ Topper
Colorado Springs, CO 80840

Commander
LCDR George Eustace
U.S. Naval School
Civil Engineering Corps, Office
ATTN: Code 061/LCDR Eustace
Port Hueneme, CA 93043

Automated Construction Management System. -- Champaign, Ill : Construction
Engineering Research Laboratory, 1984.
2 v. (Technical report ; P-158)

Contents. -- v.I. User's guide / by Jennifer S. Young. -- v.II. Program documentation / by Charles E. Herring, Jr.

1. Automated Construction Management System. 2. U.S. Army -- Military construction operations -- data processing. 3. U.S. Army, Engineer Brigade, 18th. I. Young, Jennifer S. II. Herring, Charles E. III. Series: Technical report (Construction Engineering Research Laboratory) ; P-158.

END

FILMED

8

DTIC